DIANet: Dense-and-Implicit Attention Network

Zhongzhan Huang,[1*] **Senwei Liang,**[2*] **Mingfu Liang,**[3] **Haizhao Yang**[2,4†]

[1]Tsinghua University, [2]Purdue University, [3]Northwestern University, [4]National University of Singapore[‡]
hzz_dedekinds@foxmail.com, {liang339, haizhao}@purdue.edu,
mingfuliang2020@u.northwestern.edu

Abstract

Attention networks have successfully boosted the performance in various vision problems. Previous works lay emphasis on designing a new attention module and individually plug them into the networks. Our paper proposes a novel-and-simple framework that shares an attention module throughout different network layers to encourage the integration of layer-wise information and this parameter-sharing module is referred to as Dense-and-Implicit-Attention (DIA) unit. Many choices of modules can be used in the DIA unit. Since Long Short Term Memory (LSTM) has a capacity of capturing long-distance dependency, we focus on the case when the DIA unit is the modified LSTM (called DIA-LSTM). Experiments on benchmark datasets show that the DIA-LSTM unit is capable of emphasizing layer-wise feature interrelation and leads to significant improvement of image classification accuracy. We further empirically show that the DIA-LSTM has a strong regularization ability on stabilizing the training of deep networks by the experiments with the removal of skip connections (He et al. 2016a) or Batch Normalization (Ioffe and Szegedy 2015) in the whole residual network.

Introduction

Attention, a cognitive process that selectively focuses on a small part of information while neglects other perceivable information (Anderson 2005), has been used to effectively ease neural networks from learning large information contexts from sentences (Vaswani et al. 2017; Britz et al. 2017; Cheng, Dong, and Lapata 2016), images (Xu et al. 2015; Luong, Pham, and Manning 2015) and videos (Miech, Laptev, and Sivic 2017). Especially in computer vision, deep neural networks (DNNs) incorporated with special operators that mimic the attention mechanism can process informative regions in an image efficiently. These operators are modularized and plugged into networks as attention modules (Hu, Shen, and Sun 2018; Woo et al. 2018; Park et al. 2018; Wang et al. 2018; Hu et al. 2018; Cao et al. 2019).

---

[*]Equal contribution
[†]Corresponding author
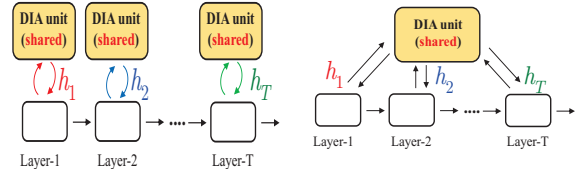[‡]Part of the work was done in Singapore

Figure 1: **Left.** Explicit structure of the network with DIA unit. **Right.** Implicit connection caused by DIA unit.

Previous works lay emphasis on designing a new attention module and individually plug them into networks. Generally, the attention module can be divided into three parts: extraction, processing and recalibration. First, the plug-in module extracts internal features of a network which can be squeezed channel-wise information (Hu, Shen, and Sun 2018; Li et al. 2019) or spatial information (Wang et al. 2018; Woo et al. 2018; Park et al. 2018). Next, the module processes the extraction and generates a mask to measure the importance of the features via a fully connected layer (Hu, Shen, and Sun 2018), convolution layer (Wang et al. 2018). Last, the mask is applied to recalibrate the features. Previous works focus on designing effective ways to process the extracted features. There is one obvious common ground where the attention modules are individually plugged into each layer throughout DNNs (Hu, Shen, and Sun 2018; Woo et al. 2018; Park et al. 2018; Wang et al. 2018).

Our Framework. Differently, we propose a novel-and-simple framework that shares an attention module throughout different network layers to encourage the integration of layer-wise information and this parameter-sharing module is referred to as Dense-and-Implicit-Attention (DIA) unit. The structure and computation flow of a DIA unit is visualized in Figure 2. There are also three parts: extraction (①), processing (②) and recalibration (③) in the DIA unit. The ② is the main module in the DIA unit to model network attention and is the key innovation of the proposed method where the parameters of the attention module are shared.

Characteristics and Advantages. (1) As shown in Figure 2, the DIA unit is placed parallel to the network backbone, and it is shared with all the layers in the same
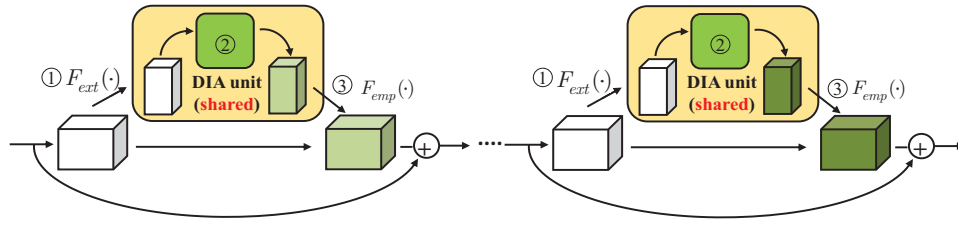
Figure 2: DIA units in the residual network. F_{ext} means the operation for extracting different scales of features. F_{emp} means the operation for emphasizing or recalibrating features.

stage (the collection of successive layers with same spatial size, as defined in (He et al. 2016a)) to improve the interaction of layers at different depth. (2) As the DIA unit is shared, the number of parameter increment from the DIA unit remains roughly constant as the depth of the network increases.

We show the feasibility of our framework by applying SE module (Hu, Shen, and Sun 2018) in the DIA unit. SE module, a representative of attention mechanism, is used for each block individually in its original design. In our framework, we share the same SE module (referred to as DIA-SE) throughout all layers in the same stage. It is easy to see that DIA-SE has the same computation cost as SE, but Table 1 shows that DIA-SE has a better generalization and smaller parameter increment.

model	#P(M)	top1-acc.
Org	1.73	$73.43_{(\pm 0.43)}$
SE	1.93	$75.03_{(\pm 0.33)}$
DIA-SE	**1.74**	$\mathbf{75.74}_{(\pm 0.41)}$

Table 1: Testing accuracy (mean $\pm$ std %) on CIFAR100 and ResNet164 with different attention modules. "Org" means the original backbone of ResNet164. #P(M) means the number of parameters (million).

Implicit and Dense Connection. We illustrate how the DIA unit connects all layers in the same stage implicitly and densely. Consider a stage consisting of many layers in Figure 1 (Left). It is an explicit structure with a DIA unit and one layer seems not to connect the other layers except the network backbone. In fact, the different layers use the parameter-sharing attention module and the layer-wise information jointly influences the update of learnable parameters in the module, which causes implicit connections between layers with the help of the shared DIA unit as in Figure 1 (Right). Since there is communication between every pair of layers, the connections over all layers are dense.

DIA-LSTM

The idea of parameter sharing is also used in Recurrent Neural Network (RNN) to capture contextual information so we consider applying RNN in our framework to model the layer-wise interrelation. Since Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber 1997) has a

strong capacity of capturing long-distance dependency, we mainly focus on the case when we use LSTM in DIA unit (DIA-LSTM) and the remainder of our paper studies DIA-LSTM.

Figure 3 is the showcase of DIA-LSTM. A global average pooling (GAP) operator (as the ① in Figure 2) is used to extract global information from current layer. A LSTM module (as the ② in Figure 2) is used to integrate multi-scale information and there are three inputs passed to the LSTM: the extracted global information from current raw feature map, the hidden state vector h_{t-1}, and the cell state vector c_{t-1} from previous layers. Then the LSTM outputs the new hidden state vector h_t and the new cell state vector c_t. The cell state vector c_t stores the information from the t-th layer and its preceding layers. The new hidden state vector h_t (dubbed as attention vector in our work) is then applied back to the raw feature map by channel-wise multiplication (as the ③ in Figure 2) to recalibrate the feature.

The LSTM in the DIA unit plays a role to bridge the current layer and the preceding layers such that the DIA unit can adaptively learn the non-linearity relationship between features in two different dimensions. The first dimension of features is the internal information of the current layer. The second dimension represents the outer information, regarded as layer-wise information, from the preceding layers. The non-linearity relationship between these two dimensions can benefit attention modeling for the current layer. The multi-dimensional modeling enables DIA-LSTM to have regularization effect.

Our Contribution

We summarize our contribution as followed,

1. We propose a novel-and-simple framework that shares an attention module throughout different network layers to encourage the integration of layer-wise information.

2. We propose incorporating LSTM in the DIA unit (DIA-LSTM) and show the effectiveness of DIA-LSTM for image classification by conducting experiments on benchmark datasets and popular networks.

Related Works

Attention Mechanism in Computer Vision. Previous works use the attention mechanism in image classification via utilizing a recurrent neural network to select and process local regions at high resolution sequentially (Mnih et al.
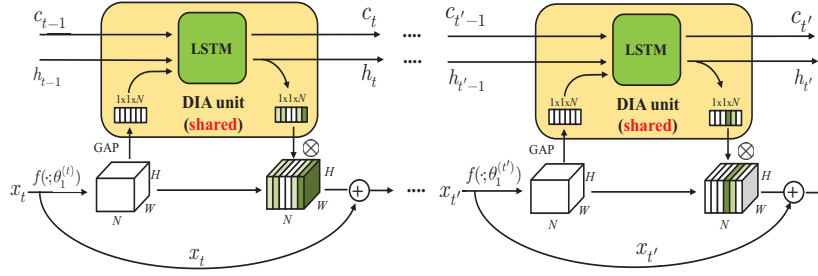
Figure 3: The showcase of DIA-LSTM. In the LSTM cell, c_t is the cell state vector and h_t is the hidden state vector. GAP means global average pool over channels and $\otimes$ means channel-wise multiplication.

2014; Zhao et al. 2017). Concurrent attention-based methods tend to construct operation modules to capture non-local information in an image (Wang et al. 2018; Cao et al. 2019), and model the interrelationship between channel-wise features (Hu, Shen, and Sun 2018; Hu et al. 2018). The combination of multi-level attention is also widely studied (Park et al. 2018; Woo et al. 2018; Wang et al. 2019; 2017). Prior works (Wang et al. 2018; Cao et al. 2019; Hu, Shen, and Sun 2018; Hu et al. 2018; Park et al. 2018; Woo et al. 2018; Wang et al. 2019) usually insert an attention module in each layer individually. In this work, the DIA unit is innovatively shared for all the layers in the same stage of the network, and the existing attention modules can be composited into the DIA unit readily. Besides, we adopt a global average pooling in part ① to extract global information and a channel-wise multiplication in part ③ to recalibrate features, which is similar to SENet (Hu, Shen, and Sun 2018).

Dense Network Topology. DenseNet proposed in (Huang et al. 2017) connects all pairs of layers directly with an identity map. Through reusing features, DenseNet has the advantage of higher parameter efficiency, a better capacity of generalization, and more accessible training than alternative architectures (Lin, Chen, and Yan 2013; He et al. 2016a; Srivastava, Greff, and Schmidhuber 2015b). Instead of explicitly dense connections, the DIA unit implicitly links layers at different depth via a shared module and leads to dense connection.

Multi-Dimension Feature Integration. (Wolf and Bileschi 2006) experimentally analyzes that even the simple aggregation of low-level visual features sampled from a wide inception field can be efficient and robust for context representation, which inspires (Hu, Shen, and Sun 2018; Hu et al. 2018) to incorporate multi-level features to improve the network representation. (Li, Ouyang, and Wang 2016) also demonstrates that by biasing the feature response in each convolutional layers using different activation functions, the deeper layer could achieve the better capacity of capturing the abstract pattern in DNN. In the DIA unit, the high non-linearity relationship between multidimensional features is learned and integrated via the LSTM module.

Dense-and-Implicit Attention Network

In this section, we formally introduce the DIA-LSTM unit. We incorporate the modified LSTM module in the DIA unit.

Afterward, a DIANet is referred to as a network built with DIA-LSTM units.

Formulation of DIA-LSTM Unit

Figure 3 shows a DIA-LSTM unit built with a residual network (He et al. 2016a), the input of the t-th layer is $x_t \in \mathbb{R}^{W \times H \times N}$, where W, H and N mean width, height and the number of channels, respectively. $f(\cdot; \theta_1^{(t)})$ is the residual mapping at the t-th layer with parameters $\theta_1^{(t)}$ as introduced in (He et al. 2016a). Let $a_t = f(x_t; \theta_1^{(t)}) \in \mathbb{R}^{W \times H \times N}$. Next, a global average pooling denoted as $\text{GAP}(\cdot)$ is applied to a_t to extract global information from features in the current layer. Then $\text{GAP}(a_t) \in \mathbb{R}^N$ is passed to LSTM along with a hidden state vector h_{t-1} and a cell state vector c_{t-1} (h_0 and c_0 are initialized as zero vectors). The LSTM finally generates a current hidden state vector $h_t \in \mathbb{R}^N$ and a cell state vector $c_t \in \mathbb{R}^N$ as

$$(h_t, c_t) = \text{LSTM}(\text{GAP}(a_t), h_{t-1}, c_{t-1}; \theta_2), \quad (1)$$

where θ_2 is the trainable parameter within the LSTM. In our model, the hidden state vector h_t is regarded as attention vector to adaptively recalibrate feature maps. We apply channel-wise multiplication $\otimes$ to enhance the importance of features, i.e., $a_t \otimes h_t$. We obtain x_{t+1} after skip connection, i.e., $x_{t+1} = x_t + a_t \otimes h_t$.

Table 2 shows the formulation of ResNet, SENet, and DIANet, and Part (b) is the main difference between them. The LSTM module is used repeatedly and shared with different layers in parallel to the network backbone. Therefore the number of parameters θ_2 in an LSTM does not depend on the number of layers in the backbone, e.g., t. SENet utilizes an attention-module consisted of fully connected layers to model the channel-wise dependency for each layer individually (Hu, Shen, and Sun 2018). The total number of parameters brought by the added-in modules depends on the number of layers in the backbone and increases concerning the number of layers.

Modified LSTM Module

Now we introduce the modified LSTM module used in Figure 3. The design of the attention module usually requires the value of the attention vector in the range $[0, 1]$

	ResNet	SENet	DIANet (ours)
(a)	$a_t = f(x_t; \theta_1^{(t)})$	$a_t = f(x_t; \theta_1^{(t)})$	$a_t = f(x_t; \theta_1^{(t)})$
(b)	-	$h_t = \text{FC}(\text{GAP}(a_t); \theta_2^{(t)})$	$(h_t, c_t) = \text{LSTM}(\text{GAP}(a_t), h_{t-1}, c_{t-1}; \theta_2)$
(c)	$x_{t+1} = x_t + a_t$	$x_{t+1} = x_t + a_t \otimes h_t$	$x_{t+1} = x_t + a_t \otimes h_t$

Table 2: Formulation for the structure of ResNet, SENet and DIANet. f is the convolution layer. $\text{GAP}(\cdot)$ means global average pooling. FC means fully connected layer and $\theta_2^{(t)}$ is the trainable parameter within SE module at the t-th layer.

and also requires a small parameter increment. We conduct some modifications in the LSTM module used in DIA-LSTM. As shown in Figure 4, compared to the standard LSTM (Hochreiter and Schmidhuber 1997) module, there are two modifications in our purposed LSTM: 1) a shared linear transformation to reduce input dimension of LSTM; 2) a carefully selected activation function for better performance.

(1) Parameter Reduction. A standard LSTM consists of four linear transformation layers as shown in Figure 4 (Top). Since y_t, h_{t-1} and h_t are of the same dimension N, the standard LSTM may cause $8N^2$ parameter increment as shown in the last section (Analysis). When the number of channels is large, e.g., $N = 2^{10}$, the parameter increment of the added-in LSTM module in the DIA unit will be over 8 million, which can hardly be tolerated.

As shown Figure 4 (Top), h_{t-1} and y_t in the standard LSTM are passed to four linear transformation layers with the same input and output dimension N. In the DIA-LSTM, a linear transformation layer (denoted as "Linear1" in Figure 4 (Bottom)) with a smaller output dimension are applied to h_{t-1} and y_t. We use reduction ratio r in "Linear1". Specifically, we reduce the dimension of the input from N to N/r and then apply the ReLU activation function to increase nonlinearity in this module. The dimension of the output of the ReLU layer is changed back to N when the output is passed to those four linear transformation functions. This modification can enhance the relationship between the inputs for different parts in DIA-LSTM and also effectively reduce the number of parameters by sharing a linear transformation for dimension reduction. The number of parameter increment reduces from $8N^2$ to $10N^2/r$ as shown in the last section (Analysis). When we choose an appropriate reduction ratio r, we can make a better trade-off between parameter reduction and the performance of DIANet. Further experimental results will be discussed in the ablation study later.

(2) Activation Function. Sigmoid ($\sigma(z) = 1/(1 + e^{-z})$) is used in many attention-based methods like SENet (Hu, Shen, and Sun 2018), CBAM (Woo et al. 2018) to generate attention maps as a gate mechanism. As shown in Figure 4 (Bottom), we change the activation function of the output layer from Tanh to Sigmoid. Further discussion will be presented in ablation study.

Experiments

In this section, we evaluate the performance of the DIA-LSTM unit in image classification and empirically demonstrate its effectiveness. We conduct experiments on popular
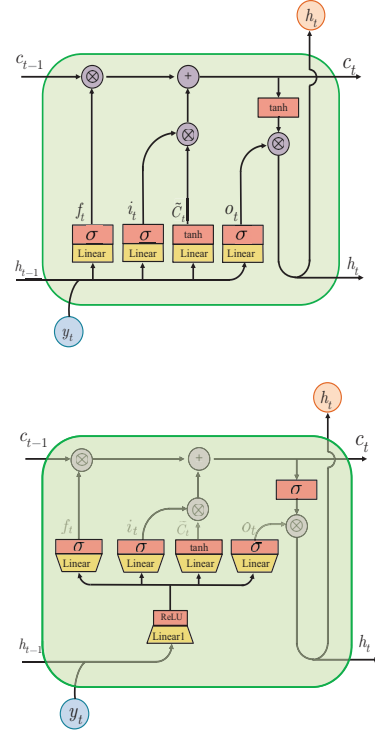

Figure 4: **Top**. The standard LSTM cell. **Bottom**. The modified LSTM cell in DIA-LSTM unit. We highlight the modified component in the modified LSTM. "σ" means the sigmoid activation. "Linear" means the linear transformation.

networks for benchmark datasets. Since SENet (Hu, Shen, and Sun 2018) is also a channel-specific attention model, we compare DIANet with SENet. For a fair comparison, we adjust the reduction ratio such that the number of parameters of DIANet is similar to that of SENet.

Dataset and Model. We conduct experiments on CIFAR10, CIFAR100 (Krizhevsky and Hinton 2009) and ImageNet 2012 (Russakovsky et al. 2015) using ResNet (He et al. 2016a), PreResNet (He et al. 2016b), WRN (Zagoruyko and Komodakis 2016) and ResNeXt (Xie et al. 2017). CIAFR10 or CIFAR100 has 50k train images and 10k test images of size 32 by 32, but has 10 and 100 classes respectively. ImageNet 2012 (Russakovsky et al. 2015) comprises 1.28 million training and 50k validation images from 1000 classes, and the random cropping of size 224 by 224 is used

	Dataset	original		SENet		DIANet		
		#P(M)	top1-acc.	#P(M)	top1-acc.	#P(M)	top1-acc.	r
ResNet164	CIFAR100	1.73	73.43	1.93	75.03	1.95	**76.67**	4
PreResNet164	CIFAR100	1.73	76.53	1.92	77.41	1.96	**78.20**	4
WRN52-4	CIFAR100	12.07	79.75	12.42	80.35	12.30	**80.99**	4
ResNeXt101,8x32	CIFAR100	32.14	81.18	34.03	82.45	33.01	**82.46**	4
ResNet164	CIFAR10	1.70	93.54	1.91	94.27	1.92	**94.58**	4
PreResNet164	CIFAR10	1.70	95.01	1.90	95.18	1.94	**95.23**	4
WRN52-4	CIFAR10	12.05	95.96	12.40	95.95	12.28	**96.17**	4
ResNeXt101,8x32	CIFAR10	32.09	95.73	33.98	96.09	32.96	**96.24**	4
ResNet34	ImageNet	21.81	73.93	21.97	74.39	21.98	**74.60**	20
ResNet50	ImageNet	25.58	76.01	28.09	76.61	28.38	**77.24**	20
ResNet152	ImageNet	60.27	77.58	66.82	78.36	65.85	**78.87**	10
ResNeXt50,32x4	ImageNet	25.03	77.19	27.56	78.04	27.83	**78.32**	20

Table 3: Testing accuracy (%) on CIFAR10, CIFAR100 and ImageNet 2012. "#P(M)" means the number of parameters (million). The rightmost "r" indicates the reduction ratio of DIANet.

in our experiments. The details can be found in Appendix.

Image Classification. As shown in Table 3, DIANet improves the testing accuracy significantly over the original networks and consistently comparing with SENet for different datasets. In particular, the performance improvement of the ResNet with the DIA unit is most remarkable. Due to the popularity of ResNet, the DIA unit may be applied in other computer vision tasks.

Ablation Study

In this section, we conduct ablation experiments to explore how to better plug DIA-LSTM units in different neural network structures and gain a deeper understanding of the role of components in the unit. All experiments are performed on CIFAR100 with ResNet. For simplicity, DIANet164 is denoted as a 164-layer ResNet built with DIA-LSTM units.
Reduction Ratio. The reduction ratio is the only hyperparameter in DIANet. The main advantage of our model is improving the generalization ability with a light parameter increment. The smaller reduction ratio causes a higher parameter increment and model complexity. This part investigates the trade-off between the model complexity and performance. As shown in Table 4, the number of parameters of the DIANets decreases with the increasing reduction ratio, but the testing accuracy declines slightly, which suggests that the model performance is not sensitive to the reduction ratio. In the case of $r = 16$, the DIANet164 has 0.05M parameter increment compared to the original ResNet164 but the testing accuracy of the DIANet164 is 76.50% while that of the ResNet164 is 73.43%.
The Depth of Networks. Generally in practice, DNNs with a larger number of parameters do not guarantee sufficient performance improvement. Deeper networks may contain extreme feature and parameter redundancy (Huang et al. 2017). Therefore, designing a new structure of deep neural networks is necessary (He et al. 2016a; Huang et al. 2017; Srivastava, Greff, and Schmidhuber 2015a; Hu, Shen, and Sun 2018; Hu et al. 2018; Wang et al. 2018). Since DIA

Ratio r	#P(M)	top1-acc.
1	$2.59_{(+0.86)}$	76.88
4	$1.95_{(+0.22)}$	76.67
8	$1.84_{(+0.11)}$	76.42
16	$1.78_{(+0.05)}$	76.50

Table 4: Test accuracy (%) with different reduction ratio on CIFAR100 with ResNet164. The value in bracket means the parameter increment compared with the original ResNet164 (1.73M).

units change the topology of DNN backbones, evaluating the effectiveness of DIANet structure is important. Here we investigate how the depth of DNNs influences DIANets in two aspects: (1) the performance of DIANets compared to SENets of various depth; (2) the parameter increment of DIANets.

The results in Table 5 show that as the depth increases from 83 to 407 layers, the DIANet with a smaller number of parameters can achieve higher classification accuracy than the SENet. Moreover, the DIANet83 can achieve a similar performance as the SENet245, and DIANet164 outperforms all the SENets with at least 1.13% and at most 58.8% parameter reduction. They imply that the DIANet is of higher parameter efficiency than SENet. The results also suggest that: for DIANet, as shown in Figure 3, the DIA-LSTM unit will pass more layers recurrently with a deeper depth. The DIA-LSTM can handle the interrelationship between the information of different layers in much deeper DNN and figure out the long-distance dependency between layers.
Activation Function and the Number of Stacking Cells. We choose different activation functions in the output layer of LSTM in Figure 4 (Bottom) and different numbers of stacking LSTM cells to explore the effects of these two factors. In Table 6, we find that the performance has been significantly improved after replacing tanh in the standard
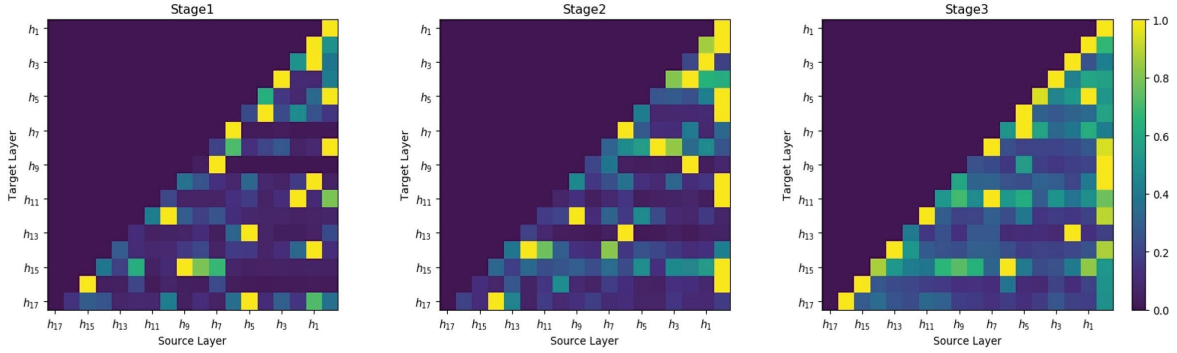
Figure 5: Visualization of feature integration for each stage by random forest. Each row presents the importance of source layers $h_n, 1 \leq n < t$ contributing to the target layer h_t.

CIFAR-100	SENet		DIANet($r = 4$)	
Depth	#P(M)	acc.	#P(M)	acc.
ResNet83	0.99	74.67	$1.11_{(+0.12)}$	75.02
ResNet164	1.93	75.03	$1.95_{(+0.02)}$	76.67
ResNet245	2.87	75.03	$2.78_{(-0.09)}$	76.79
ResNet407	4.74	75.54	$4.45_{(-0.29)}$	76.98

Table 5: Test accuracy (%) with ResNet of different depth on CIFAR100.

#P(M)	Activation	#LSTM cells	top1-acc.
1.95	sigmoid	1	76.67
1.95	tanh	1	75.24
1.95	ReLU	1	74.62
3.33	sigmoid	3	75.20
3.33	tanh	3	76.47

Table 6: Test accuracy (%) on CIFAR100 with DIANet164 of different activation function at the output layer in the modified LSTM and different number of stacking LSTM cells.

LSTM with sigmoid. As shown in Figure 4 (Bottom), this activation function is located in the output layer and directly changes the effect of memory unit c_t on the output of the output gate. In fact, the sigmoid is used in many attention-based methods like SENet as a gate mechanism. The test accuracy of different choices of LSTM activation functions in Table 6 shows that sigmoid better helps LSTM as a gate to rescale channel features. Table 12 in the SENet paper (Hu, Shen, and Sun 2018) shows the performance of different activation functions like: sigmoid > tanh > ReLU (bigger is better), which coincides to our reported results.

When we use sigmoid in the output layer of LSTM, the increasing number of stacking LSTM cells does not necessarily lead to performance improvement but may lead to performance degradation. However, when we choose tanh, the situation is different. It suggests that, through the stacking of LSTM cells, the scale of the information flow among them is changed, which may affect the performance.

Analysis

In this section, we study some properties of DIANet, including feature integration and regularization effect on stabilizing training. Firstly, the layers are connected by the DIA-LSTM unit in DIANet and we use the random forest model (Gregorutti, Michel, and Saint-Pierre 2017) to visualize how the current layer depends on the preceding layers. Secondly, we study the stabilizing training effect of DIANet by removing all the Batch Normalization (Ioffe and Szegedy 2015) or the skip connection in the residual network.

Feature Integration

Here we try to understand the dense connection from the numerical perspective. As shown in Figure 3 and 1, the DIA-LSTM bridges the connections between layers by propagating the information forward through h_t and c_t. Moreover, h_t at different layers are also integrating with $h_{t'}, 1 \leq t' < t$ in DIA-LSTM. Notably, h_t is applied directly to the features in the network at each layer t. Therefore the relationship between h_t at different layers somehow reflects layer-wise connection degree. We explore the nonlinear relationship between the hidden state h_t of DIA-LSTM and the preceding hidden state $h_{t-1}, h_{t-2}, ..., h_1$, and visualize how the information coming from $h_{t-1}, h_{t-2}, ..., h_1$ contributes to h_t. To reveal this relationship, we consider using the random forest to visualize variable importance. The random forest can return the contributions of input variables to the output separately in the form of importance measure, e.g., Gini importance (Gregorutti, Michel, and Saint-Pierre 2017). The computation details of Gini importance can be referred to Algorithm 1.

Take $h_n, 1 \leq n < t$ as input variables and h_t as output variable, we can get the Gini importance of each variable $h_n, 1 \leq n < t$. ResNet164 contains three stages, and each stage consists of 18 layers. We conduct three Gini importance computation to each stage separately. As shown in Figure 5, each row presents the importance of source layers $h_n, 1 \leq n < t$ contributing to the target layer h_t. In

Algorithm 1 Calculate feature integration using Gini importance by Random Forest

---

Input: H: composed of $h_1,h_2,...,h_t$ from Stage i;
The size of H is ($b_z \times c_z \times f_z$)
b_z denotes the number of training samples
c_z denotes the number of channels in Stage i
f_z denotes the number of layers in Stage i
Output: The hotmap G about the feature integration for Stage i;

1: initial $G = \emptyset$;
2: **for** $j = 1$ to $f_z - 1$ **do**
3:    $x \leftarrow [h_1, h_2, ..., h_{j-1}]$;
4:    $y \leftarrow [h_j]$;
5:    $x \leftarrow x.$reshape$(b_z,(f_z - j) \times c_z)$;
6:    RF $\leftarrow$ RandomForestRegressor();
7:    RF.fit(x,y);
8:    Gini_importances $\leftarrow$ RF.feature_importances_;
     # The length of Gini_importance is $(f_z - j) \times c_z$
9:    res $\leftarrow \emptyset$; $s \leftarrow 0$; cnt $\leftarrow 0$;
10:    **for** $k = 0$ to $(f_z - j)$ **do**
11:      $s \leftarrow s + $ Gini_importance(k);
12:      cnt $\leftarrow$ cnt + 1;
13:      **if** cnt == c_z-1 **then**
14:        res.add(s);$s \leftarrow 0$;cnt $\leftarrow 0$;
15:      **end if**
     G.add(res/$\max$(res));
16:    **end for**
17: **end for**

---

each sub-graph of Figure 5, the diversity of variable importance distribution indicates that the current layer uses the information of the preceding layers. The interaction between shallow and deep layers in the same stage reveals the effect of implicitly dense connection. In particular, taking h_{17} in stage 1 (the last row) as an example, h_{16} or h_{15} does not intuitively provide the most information for h_{17}, but h_5 does. The DIA unit can adaptively integrate information between multiple layers. Moreover, in Figure 5 (stage 3), the infor-

remove	#P(M)	#P(M)↓	top1-acc.	top1-acc.↓
stage1	1.94	0.01	76.27	0.40
stage2	1.90	0.05	76.25	0.42
stage3	1.78	0.17	75.40	1.27

Table 7: The test accuracy (%) of DIANet164 with the removal of DIA-LSTM unit in different stage.

mation interaction with previous layers in stage 3 is more intense and frequent than that of the first two stages. Correspondingly, as shown in Table 7, in the experiments when we remove the DIA-LSTM unit in stage 3, the classification accuracy decreases from 76.67 to 75.40. However, when it in stage 1 or 2 is removed, the performance degradation is very similar, falling to 76.27 and 76.25 respectively. Also note that for DIANet, the number of parameter increment in stage 2 is larger than that of stage 1. It implies that the significant performance degradation after the removal of stage

3 may be not only due to the reduction of the number of parameters but due to the lack of dense feature integration.

The Effect on Stabilizing Training

Removal of Batch Normalization. Small changes in shallower hidden layers may be amplified as the information propagates within the deep architecture and sometimes result in a numerical explosion. Batch Normalization (BN) (Ioffe and Szegedy 2015) is widely used in the deep networks since it stabilizes the training by normalization the input of each layer. DIA-LSTM unit recalibrates the feature maps by channel-wise multiplication, which plays a role of scaling similar to BN. Table 8 shows the performance of the models of different depth trained on CIFAR100 and BNs are removed in these networks. The experiments are conducted on a single GPU with batch size 128 and initial learning rate 0.1. Both the original ResNet, SENet face problem of numerical explosion without BN while the DIANet can be trained with depth up to 245. In Table 8, at the same depth, SENet has larger number of parameters than DIANet but still comes to numerical explosion without BN, which means that the number of parameters is not the case for stabilization of training but sharing mechanism we proposed may be the case. Besides, comparing with Table 5, the testing accuracy of DIANet without BN still can keep up to 70%. The scaling learned by DIANet integrates the information from preceding layers and enables the network to choose a better scaling for features of current layer.

Removal of Skip Connection. The skip connection has become a necessary structure for training DNNs (He et al. 2016b). Without skip connection, the DNN is hard to train due to the reasons like the gradient vanishing (Bengio et al. 1994; Glorot and Bengio 2010; Srivastava, Greff, and Schmidhuber 2015a). We conduct the experiment where all the skip connections are removed in ResNet56 and count the absolute value of gradient at the output tensor of each stage. As shown in Figure 6 which presents the gradient distribution with all skip connection removal, DIANet (blue) obviously enlarges the mean and variance of the gradient distribution, which enables larger absolute value and diversity of gradient and relieves gradient degradation to some extent.

Without Data Augment. Explicit dense connections may help bring more efficient usage of parameters, which makes the neural network less prone to overfit (Huang et al. 2017). Although the dense connections in DIA-LSTM are implicit, the DIANet still shows the ability to reduce overfitting. To verify it, We train the models without data augment to reduce the influence of regularization from data augment. As shown in Table 9, DIANet achieves lower testing error than ResNet164 and SENet. To some extent, the implicit and dense structure of DIANet may have regularization effect.

Number of Parameters in LSTM

This section shows the number of parameter costs in the standard LSTM and the modified LSTM with the reduction ratio r. The input y_t, the hidden state vector h_{t-1} and the output in Figure 4 are of the same size N, which is equal to the number of channels.

	original		SENet		DIANet($r=16$)	
	#P(M)	top1-acc.	#P(M)	top1-acc.	#P(M)	top1-acc.
ResNet83	0.88	nan	0.98	nan	0.94	**70.58**
ResNet164	1.70	nan	1.91	nan	1.76	**72.36**
ResNet245	2.53	nan	2.83	nan	2.58	**72.35**
ResNet326	3.35	nan	3.75	nan	3.41	nan

Table 8: Testing accuracy (%). We train models of different depth without BN on CIFAR-100. "nan" indicates the numerical explosion.
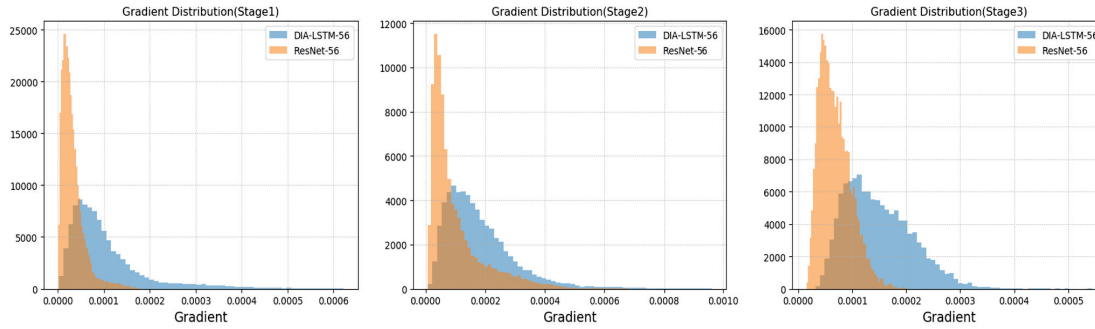
Figure 6: The distribution of gradient in each stage of ResNet56 without all the skip connections.

Models	CIFAR-10	CIFAR-100
ResNet164	87.32	60.92
SENet	88.30	62.91
DIANet	**89.25**	**66.73**

Table 9: Test accuracy (%) of the models without data augment with ResNet164.

Standard LSTM. There are four linear transformations in the standard LSTM as shown in Figure 4 (Top) to control the information flow with input y_t and h_{t-1} respectively. To simplify the calculation, the bias is omitted. Therefore, for the y_t, the number of parameters of four linear transformations is equal to $4N^2$. Similarly, the number parameters of four linear transformations for input h_{t-1} is equal to $4N^2$. The total number equals to $8N^2$.

DIA-LSTM. As shown in Figure 4 (Bottom), there is a linear transformation to reduce the dimension at the beginning, which reduces the dimension of input y_t from N to N/r. The number of parameters for the linear transformation is equal to N^2/r. Then the output will be passed to four linear transformations the same as the standard LSTM. The number of parameters of four linear transformations is equal to $4N^2/r$. Therefore, for input y_t and reduction ratio r, the total number of parameters is $5N^2/r$. Similarly, the number of parameters with input h_{t-1} is the same as that concerning the input y_t. The total number of parameters is $10N^2/r$.

Conclusion

In this paper, we proposes a novel-and-simple framework that shares an attention module throughout different network layers to encourage the integration of layer-wise information. The parameter-sharing module is called Dense-and-Implicit Attention (DIA) unit. We propose incorporating LSTM in DIA unit (DIA-LSTM) and show the effectiveness of DIA-LSTM for image classification by conducting experiments on benchmark datasets and popular networks. We further empirically show that the DIA-LSTM has a strong regularization ability on stabilizing the training of deep networks by the experiments with the removal of skip connections or Batch Normalization (Ioffe and Szegedy 2015) in the whole residual network.

Acknowledgments

References

Anderson, J. R. 2005. *Cognitive psychology and its implications*. Macmillan.

Bengio, Y.; Simard, P.; Frasconi, P.; et al. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks* 5(2):157–166.

Britz, D.; Goldie, A.; Luong, M.-T.; and Le, Q. 2017. Massive exploration of neural machine translation architectures. *arXiv preprint arXiv:1703.03906*.

Cao, Y.; Xu, J.; Lin, S.; Wei, F.; and Hu, H. 2019. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. *arXiv preprint arXiv:1904.11492*.

Cheng, J.; Dong, L.; and Lapata, M. 2016. Long short-term memory-networks for machine reading. *EMNLP 2016*.

Glorot, X., and Bengio, Y. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 249–256.

Gregorutti, B.; Michel, B.; and Saint-Pierre, P. 2017. Correlation and variable importance in random forests. *Statistics Computing* 27(3):659–678.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016a. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016b. Identity mappings in deep residual networks. In *European Conference on Computer Vision*.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Hu, J.; Shen, L.; Albanie, S.; Sun, G.; and Vedaldi, A. 2018. Gather-excite: Exploiting feature context in convolutional neural networks. In *Advances in Neural Information Processing Systems*, 9401–9411.

Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7132–7141.

Huang, G.; Liu, Z.; Weinberger, K. Q.; and van der Maaten, L. 2017. Densely connected convolutional networks. In *Computer Vision and Pattern Recognition*.

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, 448–456. JMLR.org.

Krizhevsky, A., and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical report, Citeseer.

Li, X.; Wang, W.; Hu, X.; and Yang, J. 2019. Selective kernel networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 510–519.

Li, H.; Ouyang, W.; and Wang, X. 2016. Multi-bias nonlinear activation in deep neural networks. In *International conference on machine learning*, 221–229.

Lin, M.; Chen, Q.; and Yan, S. 2013. Network in network. *arXiv preprint arXiv:1312.4400*.

Luong, M.-T.; Pham, H.; and Manning, C. D. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Miech, A.; Laptev, I.; and Sivic, J. 2017. Learnable pooling with context gating for video classification. *arXiv preprint arXiv:1706.06905*.

Mnih, V.; Heess, N.; Graves, A.; et al. 2014. Recurrent models of visual attention. In *Advances in neural information processing systems*, 2204–2212.

Park, J.; Woo, S.; Lee, J.-Y.; and Kweon, I. S. 2018. Bam: Bottleneck attention module. *arXiv preprint arXiv:1807.06514*.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* 115(3):211–252.

Srivastava, R. K.; Greff, K.; and Schmidhuber, J. 2015a. Training very deep networks. In *Advances in neural information processing systems*, 2377–2385.

Srivastava, R. K.; Greff, K.; and Schmidhuber, J. 2015b. Highway networks. *arXiv preprint arXiv:1505.00387*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wang, F.; Jiang, M.; Qian, C.; Yang, S.; Li, C.; Zhang, H.; Wang, X.; and Tang, X. 2017. Residual attention network for image classification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7794–7803.

Wang, X.; Cai, Z.; Gao, D.; and Vasconcelos, N. 2019. Towards universal object detection by domain attention. *CoRR* abs/1904.04402.

Wolf, L., and Bileschi, S. 2006. A critical view of context. *International Journal of Computer Vision* 69(2):251–261.

Woo, S.; Park, J.; Lee, J.-Y.; and So Kweon, I. 2018. Cbam: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 3–19.

Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; and He, K. 2017. Aggregated residual transformations for deep neural networks. In *Computer Vision and Pattern Recognition*.

Xu, K.; Ba, J. L.; Kiros, R.; Cho, K.; Courville, A.; Salakhutdinov, R.; Zemel, R. S.; and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, 2048–2057. JMLR.org.

Zagoruyko, S., and Komodakis, N. 2016. Wide residual networks. In *BMVC*.

Zhao, B.; Wu, X.; Feng, J.; Peng, Q.; and Yan, S. 2017. Diversified visual attention networks for fine-grained object classification. *IEEE Transactions on Multimedia* 19(6):1245–1256.