
Reading Group Dec. 11, 2019

Suman Saha (postdoc), Computer Vision Lab @ ETH Zurich
Motivation: DNN Activation Functions

- Learning activation functions to improve deep neural networks (DNNs) [1]
- Parameters in the linear components (W and b) are learned from data
- While nonlinearities are predefined, e.g. sigmoid, tanh or ReLU etc.
- Assumption – an arbitrary complex function can be approximated using any of these common nonlinear functions
- In practice, the choice of nonlinearity affects:
  - → the learning dynamics
  - → network expressive power

Motivation: Choice of Nonlinearity

• Active research area – design activation functions that enable fast training of DNN

Vanishing Gradient Problem
• Derivative of a Sigmoid Function
• ranges between 0 to 0.25

Weight Update

\[ w_{11}^{\text{new}} = w_{11}^{\text{Old}} - \eta \frac{\partial L}{\partial w_{11}^{\text{Old}}} \]

\[ \frac{\partial L}{\partial w_{11}} = \frac{\partial o_{21}}{\partial o_{11}} \cdot \frac{\partial o_{11}}{\partial w_{11}} \]

• For DNN with more layers, the gradients tend to vanish more in the lower layers
Motivation: Choice of Nonlinearity

- The rectified linear activation function (ReLU) does not saturate like sigmoidal functions
- helps to overcome the vanishing gradient problem

**Another recent activation functions**

- Maxout activation (Goodfellow et al., 2013) – computes the maximum of a set of linear functions
- Springenberg & Riedmiller (2013) replaced the max function
- Gulcehre et al. (2014) explored an activation function that replaces the max function with an $L_p$ norm
Motivation

• The type of activation function can have a significant impact on learning
• One way to explore the space of possible functions is to learn the activation function during training (Agostinelli et al., 2014)
Adaptive Piecewise Linear (APL) units

• Activation functions as a sum of hinge-shaped functions resulting in a piecewise linear activation function

\[ h_i(x) = \max(0, x) + \sum_{s=1}^{S} a_i^s \max(0, -x + b_i^s) \]

• \( S \) (the **number of hinges**) is a hyperparameter set in advance
• \( a_i^s, b_i^s \) are the **learnable parameters**, where \( i \in 1, ..., S \)
• \( a_i^s \) variables control the **slopes** of the linear segments
• \( b_i^s \) variables determine the **locations** of the hinges
Adaptive Piecewise Linear (APL) units

\[ h_i(x) = \max(0, x) + \sum_{s=1}^{S} a_i^s \max(0, -x + b_i^s) \]

- Fig. 1 shows example APL functions for \( S = 1 \)
- for large enough \( S \), APL can approximate arbitrarily complex continuous functions
- the first term in Eq. (1) is ReLU
- when \( x < 0 \) the derivative of ReLU is 0 resulting dead neurons

Leaky ReLUs addresses the dead neurons problems, e.g. leaky ReLU may have \( y = 0.01x \) when \( x < 0 \)

Figure 1: Sample activation functions obtained from changing the parameters. Notice that figure b shows that the activation function can also be non-convex.

Figure 2
Categories of deep learning MTL: (a) Hard-sharing; (b) Soft-sharing; (c) Task Adaptive Activation Network (proposed model); (d) Inner Structure of Adaptive Activation Layer.

- **Proposed approach** = hard-sharing + learnable task-specific activation functions
- all tasks can share their weights and biases on the hidden layers
- more **scalable** than the soft-sharing methods where the number of network components is proportional to the number of tasks
TAAN

• For a task $t$, given the input from either the previous layer or data input, the output of the $l$-th AAL (Adaptive Activation Layer) is defined by

$$h_i^t = \mathcal{F}_i^t(W_i h_{l-1} + b_l),$$

• weight and bias parameters are shared across tasks

• The task-specific activation function for task $t$ and layer $l$ is defined as

$$\mathcal{F}_i^t(\cdot) = \sum_{i=1}^{M} \alpha_i^t(i) B_i(\cdot),$$

• Recall from slide 6 and 7, $M$ (the number of hinges) is a hyperparameter set in advance

• $\alpha_i^t = [\alpha_i^t(1), \cdots, \alpha_i^t(M)] \in \mathbb{R}^M$ denotes the coordinates of the basis functions $\{B_i\}_{i=1}^{M}$
TAAN

- Each task has its own coordinate vector \( \alpha_t^i \in \mathbb{R}^M \)
- There is a coordinate matrix \( \alpha_l \in \mathbb{R}^{T \times M} \) attached to each AAL hidden layer
- The coordinate matrices of the hidden layers control the level of network sharing among multiple tasks
- For instance, if tasks 1 and 2 have more shared knowledge at the 1\(^{st}\) hidden layer, \( \mathcal{F}_1^1(\cdot) \) and \( \mathcal{F}_2^1(\cdot) \) have higher similarity,
- Thus \( \alpha_1^1 \) and \( \alpha_2^1 \) are more similar
- On the other hand, if tasks 1 and 2 share less knowledge at the 2nd hidden layer, their activation functions are more diverse
- During the training phase, the coordinate matrices of all hidden layers are optimized to extract both the shared and task-specific knowledge from data
Metrics for Activation Functions

• In order to understand how TAAN captures the relationship of multiple tasks, we need a metrics to measure the difference/similarity between two activation functions
• As the basis functions of APL units are unbounded and non-orthonormal, the coordinate vectors do not reveal much property about the functions
• Besides, the commonly used $L^2$ norm and inner product are infinite almost everywhere, it is impossible to use them as metrics
• Authors redefine the finite inner product and norm assuming $X$ is a random variable with Gaussian distribution $p(x) = \mathcal{N}(\mu, \sigma^2)$
Functional Regularization

• For each layer of TAAN, the coordinate matrix $\alpha_l \in \mathbb{R}^{T \times M}$ can be learned directly from the training data
• As the tasks in MTL are generally considered to be related, it is reasonable to encourage sharing more than splitting
• This insight is incorporated into TAAN by introducing regularization term on $\alpha_l$ during training
• Authors propose two functional regularization methods to further enhance the performances of TAAN
**Functional Regularization**

**Baseline: Trace-Norm**

- The first regularization hypothesis is that the matrix $\alpha_l \in \mathbb{R}^{T \times M}$ is low-rank, as the tasks in MTL often have high correlation.
- Thus, authors introduce a regularization term to $\alpha_l$

$$\mathcal{L}_{tn}(\alpha_l) = \text{trace}(\sqrt{\alpha_l \alpha_l^T})$$

- $\sqrt{\cdot}$ denotes the square root of matrix

**Functional regularization by cosine similarity**

the similarity of two task-specific activation functions can be defined by the cosine similarity, which is computed as:

$$\mathcal{L}_{cos}(\alpha_l) = -\frac{1}{T^2} \sum_{ij} c_{ij}(\alpha_l),$$

$$c_{ij}(\alpha_l) = \frac{\langle \mathcal{F}_i^i, \mathcal{F}_i^j \rangle}{\sqrt{\langle \mathcal{F}_i^i, \mathcal{F}_i^i \rangle \langle \mathcal{F}_i^j, \mathcal{F}_i^j \rangle}},$$
Functional Regularization

Functional regularization by distance

- Given the coordinate matrix for the l-th layer of network, authors compress the distance function Eq. (3) between task-specific activation functions with the following regularization

\[ \mathcal{L}_{dis}(\alpha_l) = \frac{1}{T^2} \sum_{i,j} \mathcal{D}_{ij}(\alpha_l), \quad \mathcal{D}_{ij}(\alpha_l) = d^2(\mathcal{F}_l^i, \mathcal{F}_l^j). \]

- The training loss of a TAAN with L task-specific activation layers becomes

\[ \mathcal{L}_{\text{total}} = \sum_{t=1}^{T} \mathcal{L}_t(\mathcal{M}_t, \{x_i^t, y_i^t\}) + c \sum_{l=1}^{L} \mathcal{L}_{\text{MTL}}(\alpha_l), \]

- Where \( \mathcal{L}_{\text{MTL}} \in \{ \mathcal{L}_{tn}, \mathcal{L}_{dis}, \mathcal{L}_{cos} \} \) and c is the regularization coefficient.
Functional Regularization

Functional regularization by distance

• Given the coordinate matrix for the l-th layer of network, authors compress the distance function Eq. (3) between task-specific activation functions with the following regularization

\[ \mathcal{L}_{dis}(\alpha_l) = \frac{1}{T^2} \sum_{i,j} D_{ij}(\alpha_l), \quad D_{ij}(\alpha_l) = d^2(F_i^l, F_j^l). \]

• the training loss of a TAAN with L task-specific activation layers becomes

\[ \mathcal{L}_{total} = \sum_{t=1}^{T} \mathcal{L}_t(M_t, \{x_i^t, y_i^t\}) + c \sum_{l=1}^{L} \mathcal{L}_{MTL}(\alpha_l), \]

• Where \( \mathcal{L}_{MTL} \in \{\mathcal{L}_{tn}, \mathcal{L}_{dis}, \mathcal{L}_{cos}\} \) and c is the regularization coefficient
Multi-Domain Multi-Label Classification

• Conduct experiments on Youtube-8M, a large dataset that consists of over 6.1 billion of Youtube videos. Each video has multiple labels from a vocabulary of 3800 topical entities, which can be further grouped into 24 top-level categories.

• To create an MTL experiment, authors consider each top-level category as a specific domain.

• For each domain, they have to define a multi-label classifier to recognize various attributes of the data.
Experiments

Multi-Domain Multi-Label Classification

- The task IDs and their corresponding domains
# Experiments

**Multi-Domain Multi-Label Classification**

Table 1: The number of network parameters and classification performance on the Youtube-8M dataset

<table>
<thead>
<tr>
<th>Benchmarks</th>
<th># of params</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>Mean mAP%10</th>
</tr>
</thead>
<tbody>
<tr>
<td>STL</td>
<td>56.1 M</td>
<td>.683</td>
<td>.807</td>
<td>.835</td>
<td>.689</td>
<td>.781</td>
<td>.431</td>
<td>.559</td>
<td>.831</td>
<td>.596</td>
<td>.779</td>
<td>.551</td>
<td>.916</td>
<td>.505</td>
<td>.383</td>
<td>.699</td>
<td>.759</td>
<td>0.675</td>
</tr>
<tr>
<td>Hard-Share</td>
<td>6.93 M</td>
<td>.770</td>
<td>.838</td>
<td>.870</td>
<td>.806</td>
<td>.824</td>
<td>.598</td>
<td>.676</td>
<td>.846</td>
<td>.754</td>
<td>.808</td>
<td>.716</td>
<td>.940</td>
<td>.629</td>
<td>.680</td>
<td>.806</td>
<td>.839</td>
<td>0.775</td>
</tr>
<tr>
<td>Soft-Order</td>
<td>6.93 M</td>
<td>.785</td>
<td>.845</td>
<td>.877</td>
<td>.819</td>
<td>.825</td>
<td>.611</td>
<td>.683</td>
<td>.848</td>
<td>.764</td>
<td>.809</td>
<td>.727</td>
<td>.943</td>
<td>.648</td>
<td>.694</td>
<td>.819</td>
<td>.838</td>
<td>0.783</td>
</tr>
<tr>
<td>MRN_t</td>
<td>56.1 M</td>
<td>.664</td>
<td>.806</td>
<td>.829</td>
<td>.608</td>
<td>.778</td>
<td>.413</td>
<td>.502</td>
<td>.828</td>
<td>.576</td>
<td>.775</td>
<td>.542</td>
<td>.916</td>
<td>.383</td>
<td>.355</td>
<td>.687</td>
<td>.695</td>
<td>0.647</td>
</tr>
<tr>
<td>MRN_{full}</td>
<td>61.6 M</td>
<td>.670</td>
<td>.804</td>
<td>.828</td>
<td>.607</td>
<td>.777</td>
<td>.416</td>
<td>.506</td>
<td>.828</td>
<td>.578</td>
<td>.775</td>
<td>.543</td>
<td>.916</td>
<td>.383</td>
<td>.355</td>
<td>.688</td>
<td>.696</td>
<td>0.648</td>
</tr>
<tr>
<td>DMTRL-Tucker</td>
<td>39.1 M</td>
<td>.660</td>
<td>.817</td>
<td>.812</td>
<td>.651</td>
<td>.794</td>
<td>.473</td>
<td>.490</td>
<td>.793</td>
<td>.569</td>
<td>.742</td>
<td>.594</td>
<td>.944</td>
<td>.475</td>
<td>.475</td>
<td>.697</td>
<td>.710</td>
<td>0.669</td>
</tr>
<tr>
<td>DMTRL-LAF</td>
<td>20.1 M</td>
<td>.851</td>
<td>.869</td>
<td>.890</td>
<td>.875</td>
<td>.841</td>
<td>.691</td>
<td>.757</td>
<td>.873</td>
<td>.839</td>
<td>.854</td>
<td>.809</td>
<td>.948</td>
<td>.746</td>
<td>.792</td>
<td>.875</td>
<td>.896</td>
<td>0.838</td>
</tr>
<tr>
<td>DMTRL-TT</td>
<td>46.2 M</td>
<td>.860</td>
<td>.874</td>
<td>.893</td>
<td>.888</td>
<td>.846</td>
<td>.710</td>
<td>.778</td>
<td>.879</td>
<td>.849</td>
<td>.860</td>
<td>.822</td>
<td>.950</td>
<td>.770</td>
<td>.807</td>
<td>.881</td>
<td>.903</td>
<td>0.848</td>
</tr>
<tr>
<td><strong>TAAN (Ours)</strong></td>
<td><strong>6.93 M</strong></td>
<td><strong>.882</strong></td>
<td><strong>.896</strong></td>
<td><strong>.910</strong></td>
<td><strong>.902</strong></td>
<td><strong>.857</strong></td>
<td><strong>.739</strong></td>
<td><strong>.804</strong></td>
<td><strong>.879</strong></td>
<td><strong>.859</strong></td>
<td><strong>.828</strong></td>
<td><strong>.808</strong></td>
<td><strong>.934</strong></td>
<td><strong>.717</strong></td>
<td><strong>.774</strong></td>
<td><strong>.833</strong></td>
<td><strong>.848</strong></td>
<td><strong>0.842</strong></td>
</tr>
<tr>
<td>TAAN + $\mathcal{L}_{tn}$</td>
<td>6.93 M</td>
<td>.741</td>
<td>.824</td>
<td>.861</td>
<td>.779</td>
<td>.813</td>
<td>.572</td>
<td>.636</td>
<td>.831</td>
<td>.711</td>
<td>.785</td>
<td>.681</td>
<td>.937</td>
<td>.581</td>
<td>.637</td>
<td>.783</td>
<td>.809</td>
<td>0.749</td>
</tr>
<tr>
<td>TAAN + $\mathcal{L}_{cos}$</td>
<td>6.93 M</td>
<td>.896</td>
<td>.906</td>
<td>.915</td>
<td>.915</td>
<td>.859</td>
<td>.769</td>
<td>.830</td>
<td>.885</td>
<td>.879</td>
<td>.843</td>
<td>.828</td>
<td>.937</td>
<td>.756</td>
<td>.805</td>
<td>.854</td>
<td>.876</td>
<td><strong>0.860</strong></td>
</tr>
<tr>
<td>TAAN + $\mathcal{L}_{dis}$</td>
<td>6.93 M</td>
<td>.889</td>
<td>.899</td>
<td>.912</td>
<td>.910</td>
<td>.859</td>
<td>.752</td>
<td>.820</td>
<td>.886</td>
<td>.873</td>
<td>.836</td>
<td>.821</td>
<td>.933</td>
<td>.731</td>
<td>.789</td>
<td>.845</td>
<td>.863</td>
<td><strong>0.851</strong></td>
</tr>
</tbody>
</table>
Experiments

Multi-Domain Multi-Label Classification

<table>
<thead>
<tr>
<th>Model</th>
<th>training time</th>
<th>inference time</th>
</tr>
</thead>
<tbody>
<tr>
<td>$MRN_t$</td>
<td>156.1 s</td>
<td>8.83 s</td>
</tr>
<tr>
<td>$MRN_{full}$</td>
<td>157.6 s</td>
<td>8.83 s</td>
</tr>
<tr>
<td>DMTRL-Tucker</td>
<td>66.02 s</td>
<td>1.53 s</td>
</tr>
<tr>
<td>DMTRL-LAF</td>
<td>24.96 s</td>
<td>0.30 s</td>
</tr>
<tr>
<td>DMTRL-TT</td>
<td>52.75 s</td>
<td>1.44 s</td>
</tr>
<tr>
<td>TAAN</td>
<td>20.05 s</td>
<td>0.88 s</td>
</tr>
<tr>
<td>TAAN + $L_{tn}$</td>
<td>22.86 s</td>
<td>0.88 s</td>
</tr>
<tr>
<td>TAAN + $L_{cos}$</td>
<td>40.34 s</td>
<td>0.88 s</td>
</tr>
<tr>
<td>TAAN + $L_{dis}$</td>
<td>38.35 s</td>
<td>0.88 s</td>
</tr>
</tbody>
</table>

Table 2: Training and inference speed evaluation on Youtube-8M (Note: the dimension of hidden layer is 512)
Experiments

Visualization

- TAAN is able to capture the complicated knowledge sharing for the tasks on the Youtube-8M dataset. For instance, domain “Food & Drink” shares all the hidden layers with domain “Home & Garden”. TAAN also discovered that the domains “Food & Drink” and “Internet & Telecom” are the most unrelated, as the distances between their activation functions are always high.

Distance matrices of the activation functions in TAAN. Light colors denote less similarity.
Questions?
Thank you for your attention!
References

Orthonormal Set of Functions

A set of functions \( f_1(x), f_2(x), \ldots, f_n(x), \ldots \) is said to be orthonormal over \((a, b)\) if

\[
\int_{a}^{b} f_m(x) \cdot f_n(x) \, dx \begin{cases} = 0 & \text{if } m \neq n \\ = 1 & \text{if } m = n \end{cases}
\]