Related Weights in Cross-stitch Network for Multi-task Learning

Sudipta Bhuin
Carnegie Mellon University
Pittsburgh, PA 15213
sbhuin@andrew.cmu.edu

Parth Chadha
Carnegie Mellon University
Pittsburgh, PA 15213
pchadha@andrew.cmu.edu

George Tan
Carnegie Mellon University
Pittsburgh, PA 15213
georget1@andrew.cmu.edu

1 Introduction

Recently, deep convolutional networks have shown manifold performance improvement over other type of networks. The key reason is the inherent feature sharing over different categories. This proves that deliberately sharing features among different, but related, categories can enhance the performance of varieties of tasks, for example segmentation attribute classification and even surface normal prediction.

Multi-task learning is a subtype of transfer learning with an approach to learn related tasks within a similar domain, as an inductive bias, to improve generalization. By sharing domain information, related tasks are learned together in parallel and achieve better performance over tasks learned in isolation, especially when the labels of the target task are scarce.

This conventional approach faces problems when the relationship between tasks can not be predetermined because of the lack of specific domain knowledge or due to a high number of tasks. Basically there is no insights or theoretical principals in how much of the representation to share among tasks. This problem has motivated recent Deep Learning methods to learn the relationship between the shared and task-specific representations.

This work investigates multi-task learning on different networks to improve the performance of facial landmark detection by using attribute classification as auxiliary tasks in conjunction. We have applied three different network architectures on the facial landmark detection task and compared with the baseline of single task facial landmark detection. Additionally, we have proposed a weight regularization method adapted from multi-domain learning and show that it improves the performance of the split and cross-stitch network without much fine tuning of parameters. We intend to find a more generalized approach towards multi-task learning in Convolutional Neural Networks (CNN).

2 Related Works

Multi-task learning spans a broad scope in machine learning e.g. computer vision [1], genomic [2], natural language processing [3]. This includes representation learning and selection [4], transfer learning [5], etc. Though multi-task learning has been used in different forms in different application, for this paper, we will consider multi-task learning in the context of CNNs used in computer vision. Multi-task learning in CNNs has been used to model related task in a joint manner for example semantic segmentation and surface normal prediction[1], pose estimation and action recognition [6], facial landmark detection and attribute classification [7], auxiliary tasks detection [8] etc.

Even though CNNs learn representations shared across different categories of a task, it is still difficult to design architectures using the same representation to perform different tasks. Sharing representations between multiple tasks in CNNs improves the performance [9], however there exists a problem with which layers to transfer. Depending on the problem at hand, there is a spectrum of possible way of sharing tasks by splitting the CNN at different layers.

Existing multi-task learning architectures experiment with varying the number of layers shared between tasks and hence the network architectures differ significantly with different tasks. In general, different level of sharing works best for different tasks [1]. This problem creates a need for a more generalized model which can work for different types of tasks efficiently.

Recently, a cross-stitch unit was introduced in CNNs [1]. Conventionally, multiple layers were shared among networks of two tasks, assuming all these layers to contribute equally in both of the tasks. A cross stitch unit combines representations from multiple networks via a learnable matrix. This unit learns how much to share among related tasks without assuming any prior relationship between the tasks.

Additional recent work motivates representation sharing by learning the tasks relationship. The key idea behind this is that networks for related tasks should have similar representation in the intermediate layers and their weight should be related. This includes a model that reduces the differences in weights between similar tasks and learns a task relationship matrix [10]. Similarly, work on domain adaptation [11], another subtype of transfer learning, models the relatedness of two different domains with a regularization on the distance between the weights of the layers for each domain. The proposed method applies a well designed loss function to prevent the weight wandering too far away from each other, thus convergence can be achieved. In this paper, to show the effectiveness of adding weight regularization on a network with cross-stitch, we have performed two similar tasks of facial landmark detection and attribute classification over the facial landmark dataset [7] with different network architectures and compared the results to a single task network.

3 Problem Formulation

3.1 Split Network

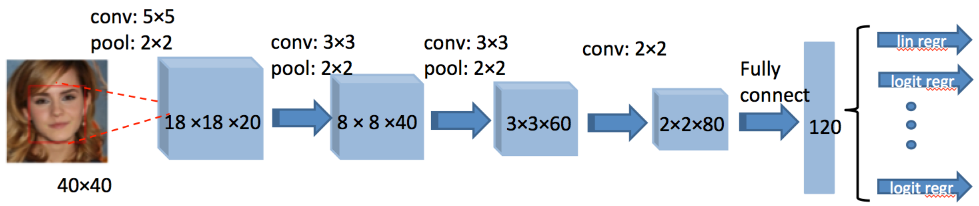


Figure 1: Split network model

We adopt the split network architecture from [7], assuming facial landmark detection as the main task and the attribute classification are auxiliary tasks. The goal is to optimize the main task which is facial landmark detection. Let the auxiliary tasks be $a \in A$ where A is the set of all the auxiliary tasks. For this problem the problem can be formulated as

$$\min \sum_{i=1}^{N} l^r(y_i^r, f(x_i; \mathbf{W}^r)) + \sum_{i=1}^{N} \sum_{a \in A} \lambda^a l^a(y_i^a, f(x_i; \mathbf{W}^a)) \tag{1}$$

where ℓ^r is the loss function of the main task, ℓ^a is the loss functions of auxiliary tasks, and λ^a is the scaling factor corresponding to the a-th task. The scaling factor denotes the importance of that particular auxiliary task. This loss function allows joint representation of linear regression and classification error. $\{x_i\}_{i=1}^N$ is the shared input feature map for N samples and the corresponding output labels are $\{y_i^r, y_i^p, y_i^g, y_i^w, y_i^s\}$. $\{y_i^r\}$ is the regression output that denotes facial landmark and $\{y_i^p, y_i^g, y_i^w, y_i^s\}$ are attribute classification of set A. The regression output $y_i^r \in R^{10}$ is the 2 dimensional coordinates of five facial landmarks: left eye, right eye, nose, left mouth corner and right mouth corner, $y_i^g, y_i^w, y_i^s \in \{0, 1\}$ are binary output representing gender, wearing glass and smiling respectively, and $y_i^p \in \{0, 1, 2, 3, 4\}$ denotes five possible output for pose: $\{0°, \pm 30°, \pm 60°\}$. The loss function for linear regression is least mean square and for the attribute classification cross entropy loss function is used. The cost function can be represented as

$$\min \frac{1}{N}\sum_{i=1}^{N} \|y_i^r - f(x_i;W^r)\|^2 - \sum_{i=1}^{N}\sum_{a\in A} \lambda^a \log(p(y_i^a|x_i;W^a)) \tag{2}$$

where $f(x_i;W^T) = (W^r)^T x$ is a linear function and the second term is a softmax function $p(y_i = m|x_i) = \frac{\exp\{(\mathbf{W}^a_m)^T x_i\}}{\sum\limits_j exp\{(\mathbf{W}^a_j)^T x_i\}}$

3.2 Cross-stitch Network

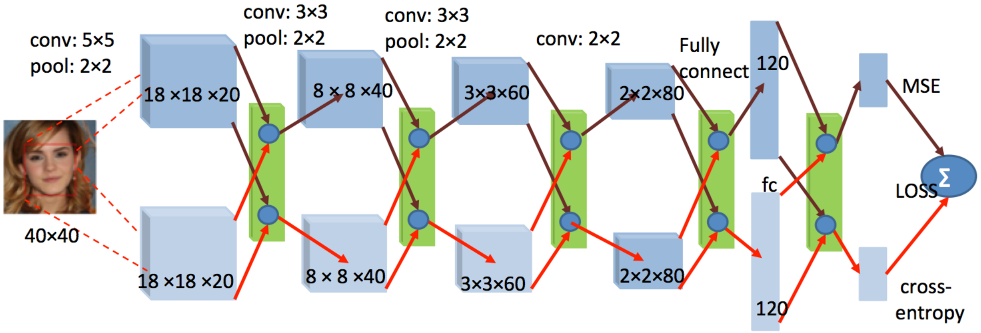


Figure 2: Cross-stitch network model

In a traditional multi-task learning framework, a joint loss is defined over all the task and the parameters are learned from that loss. However we define two separate CNN's for the two tasks and add cross-stitch units (Eq 3) which share the representation among the tasks [1].

$$\begin{pmatrix} \widetilde{\mathbf{x}}_A^{ij} \\ \widetilde{\mathbf{x}}_B^{ij} \end{pmatrix} = \begin{pmatrix} \alpha_{AA} & \alpha_{AB} \\ \alpha_{BA} & \alpha_{BB} \end{pmatrix} \begin{pmatrix} \mathbf{x}_A^{ij} \\ \mathbf{x}_B^{ij} \end{pmatrix} \tag{3}$$

The sharing is modelled as a linear combination of the activation map $\left(x_A^{ij}, x_B^{ij}\right)$ of two networks where $\left(\widetilde{x}_A^{ij}, \widetilde{x}_B^{ij}\right)$ is fed to the next layer. The linear combination parameter is learnable through back propagation. The backpropagation can be modelled as follows

$$\begin{bmatrix} \frac{\partial L}{\partial x_A^{ij}} \\ \frac{\partial L}{\partial x_B^{ij}} \end{bmatrix} = \begin{pmatrix} \alpha_{AA} & \alpha_{BA} \\ \alpha_{AB} & \alpha_{BB} \end{pmatrix} \begin{bmatrix} \frac{\partial L}{\partial \widetilde{x}_A^{ij}} \\ \frac{\partial L}{\partial \widetilde{x}_B^{ij}} \end{bmatrix} \tag{4}$$

Hence, the networks learns how much information to share among streams of different tasks by changing α [1].

3.3 Weight Regularization

We also propose a indirect weight sharing with or without the presence of cross stitch. Weight sharing is mainly popular for multi domain tasks [10] where inputs are different but the tasks are quite similar which makes us believe that the weights might be similar. Our proposal is mutual regularization the weights of the network in order to share representations indirectly. To regularize the weight of the network, we propose another cost function L_w which will be added to this network while training.

$$L = L_{task1} + L_{task1} + L_w \tag{5}$$

$$L_w = \lambda_w \sum_{j\in\Omega} r_w(W_j^{task1}, W_j^{task2}) \tag{6}$$

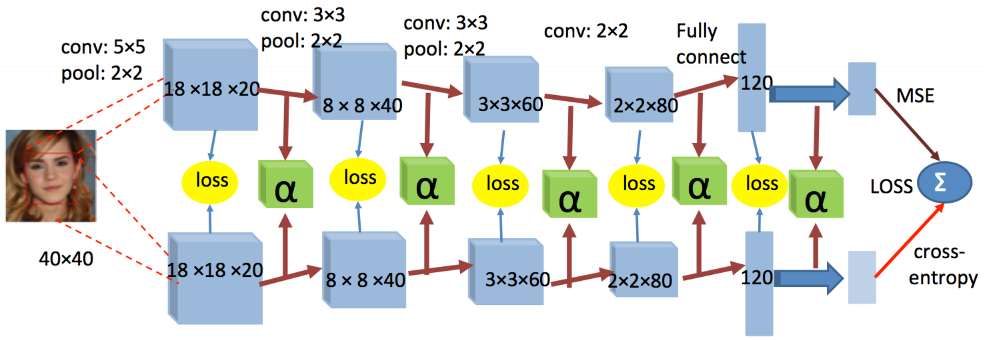


Figure 3: Cross-stitch network with weight regularization model

where L_{task1} and L_{task2} are cost function for the two tasks, L_w is mutual loss function for the weight in two streams. The weight loss can be L_2 norm, $r_w(W_j^{task1}, \theta_j^{task2}) = \left\|W_j^{task1} - W_j^{task2}\right\|^2$, however an L_2 norm penalize a small difference among weights which harms the multi-task learning if the tasks are not very related. An exponential loss can add more flexibility while still maintaining closeness of the weights. In that case, the loss function can be expressed as $r_w(W_j^{task1}, \theta_j^{task2}) = \exp(\left\|W_j^{task1} - W_j^{task2}\right\|^2) - 1$. However, we add an extra layer of learnability or flexibility to this loss function by performing a linear transformation on the weights. The final loss function can be expressed as

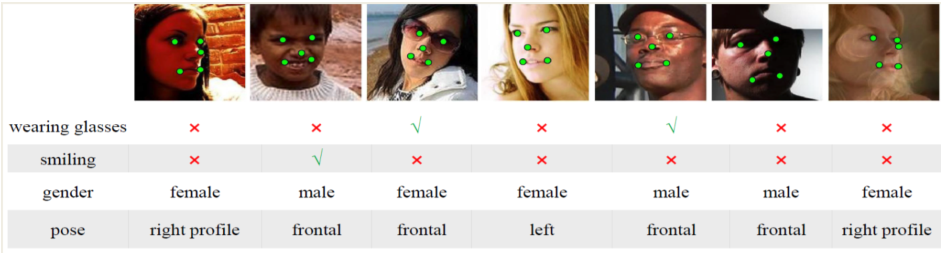
$$r_w(W_j^{task1}, \theta_j^{task2}) = \exp(\left\|a_j W_j^{task1} + b_j - W_j^{task2}\right\|^2) - 1 \tag{7}$$

where a_j and b_j are learnable scalar parameters and are different for each layer $(j \in \Omega)$. These parameters are learned while training the network. We formulate the overall loss function for training in detail in the next section where we discuss the dataset.

4 Dataset

We discuss the facial landmark dataset and formulation of loss function for this dataset for the split network, cross-stitch and cross-stitch with weight regularizer.

4.1 Facial Landmark Detection and Attribute Prediction



wearing glasses	×	×	√	×	√	×	×
smiling	×	√	×	×	×	×	×
gender	female	male	female	female	male	male	female
pose	right profile	frontal	frontal	left	frontal	frontal	right profile

Figure 4: Multi-task facial landmark detection dataset[7]

Facial landmark is related to the attribute classification. For example, the pose of the face greatly affects the facial landmark. We believe facial landmark detection [7] can be facilitated by facial attribute (smiling, gender, glasses, pose) classification through multi-task learning. We use the dataset from [7]. Data annotation has five facial landmark coordinates and four attributes of gender, smiling, wearing glasses, and head position. [7] considers facial landmark detection (FLD) as a main task but lets the network train on facial attribute classification task alongside by splitting the last layer into linear regression for FLD and logistic regression for the other attributes. The attribute classification act as auxiliary tasks.

A task constrained loss function was formulated to allow the errors of related task to be back propagated jointly. However one potential problem might be deterioration of FLD or classification due to over sharing of feature representations. Besides, the methodology suffers from lack of convergence due to the early convergence of auxiliary tasks. If the auxiliary tasks converge earlier than the main task of facial landmark regression the other networks start to overfit the main task. We propose two separate CNN chain and apply cross-stitch to share information among them. As currently cross stitch has been developed for two tasks, we consider FLD as one task and attribute classification combining all the 4 as second task and we apply cross-stitch to correlate the two tasks. We will also consider training them with different combinations of attributes.

Our approach differs from the original MTFL [7] paper by the fact that we are making the correlation among weight more flexible with cross-stitch. We hope that FLD task will not be hampered much by the other tasks and other tasks will also benefit from FLD. To optimize the FLD main task and attribute classification tasks ($a \in A$), in our proposed architecture, the loss functions can be formulated as

$$\min \frac{1}{N}\sum_{i=1}^{N} \|y_i^r - f(x_i; W^r)\|^2 - \sum_{i=1}^{N}\sum_{a\in A} \lambda^a \log(p(y_i^a|x_i; W^a)) \tag{8}$$

where $f(x_i; \mathbf{W}^a)$ is the linear term for FLD task and the second task is softmax function $p(y_i = m|x_i) = \frac{\exp\{(\mathbf{W}^a_m)^T x_i\}}{\sum_j exp\{(\mathbf{W}^a_j)^T x_i\}}$. Here $\{x_i\}_{i=1}^N$ is the input image and $\{y_i^r, y_i^p, y_i^g, y_i^w, y_i^s\}$ are corresponding labels of landmark detection and attributes: 'pose', 'gender', 'wear glasses', and 'smiling'. The training dataset available from [7] consists of 10000 annotated images.

4.2 Evaluation

Here we discuss our proposed evaluation methodology for face landmark and attribute classification tasks pair. As for the error metric for the FLD task we follow a similar metric as in [7]. The mean error is measured by the Euclidean distance of the predicted landmark from ground truth, normalized by the boundary box width. If the mean error is above 5%, it will be reported as failure. We will be evaluating our final method with the baseline comparison against single task FLD. This would verify whether the FLD accuracy is improved or remains same by applying multi task methodology. We also intend verify the improvement due to indirect weight relating with cross stitch. A more extensive fine tuning of the parameters, and learning rate can be done to improve the absolute performance of each method.

5 Design Decision

We design three different types of multi-task network architecture on Tensorflow. The input is a 40×40 image. A boundary box regression is applied to crop the image from the dataset. The 10,000 annotated images are used to train the networks from scratch. Each image is annotated with five facial landmark, i.e. left eye, right eye, nose, left mouth corner and right mouth corner and four attribute classifications, gender, glass, smiling and pose. The attributes gender, smiling and glasses each have two possible values (boy or girl, etc.). Pose has five possible values: 0, $\pm 30^\circ$ and $\pm 60^\circ$. The feature extraction stage consists of 4 convolution layers, 3 pooling layers and a fully connected layer.

5.1 Split Network

Conventional multi-task learning implies splitting the network at some layer depending on the task at hand. The decision at which layer to split is made by multiple simulations and observing in which case the performance improves. The state-of-the-art facial landmark detection with facial attribute classification uses a network split at the last fully connected layer as their initial architecture. We adopted the same architecture and similarly split at the last layer. The last fully connected layer is used in linear regression for facial landmark detection and in logistic regression for attribute classification. Intuitively every auxiliary attribute classification tasks have different levels of difficulty and have different effect on the facial landmark prediction. For example, predicting wearing glasses is easier then prediction whether a person is smiling or not. Also, a closer look at the dataset shows that, except gender, all the attributes have a mismatched positive-negative sample ratio. We use five variations of the split architecture where we combine the main task with a different auxiliary attribute classification task. We combine FLD with pose, gender, glass, smile, and all separately. Due to uncertainty of the effect of the auxiliary task and fast convergence, the loss contributed by the attribute classification is scaled down to give more importance to the main landmark regression task.

5.2 Cross-stitch Network

Two exact same networks are used for cross-stitch where each network is trained for main task and auxiliary task with cross stitch units added after every layer. Following [1], we maintain one cross stitch unit per channel. For example, in pool1, we have 16 cross-stitch units for the 16 channels. However, this also requires extensive experimentation based on the ease of convergence e.g. in [1] the surface normal prediction and semantic segmentation works better by maintaining one cross stitch unit per channel however for object detection and attribute classification only one cross-stitch unit per layer was maintained to stabilize the learning process. Each channel has extra four learnable hyperparameters α.

The training objective function is similar to Eq 8 where the first network has least mean square error function and the second network has cross entropy loss function. Similar to split architecture, five variantions of cross-stitch network is used, namely FLD+pose, FLD+gender, FLD+smile, FLD+glass and FLD+all. We initialize α_S as 0.9 and $alpha_D$ as 0.1 where α_S and α_D represents the cross-stitch parameters of same layer and different layer, respectively, i.e. $\alpha_S = \alpha_{AA} = \alpha_{BB} = 0.9$ and $\alpha_D = \alpha_{AB} = \alpha_{BA} = 0.1$.

5.3 Weight regularized Network with Cross-stitch

In the third type of architecture, weight regularization is added with the total loss function as mentioned in Eq 6. The total loss is jointly backpropagated through both the networks while training. Also, similar to first two type of architectures mentioned above, five variants of it are designed. Note that in all the three cases the loss functions are scaled lower to give more importance on the main landmark regression task.

6 Results and Analysis

6.1 Split Network predictions

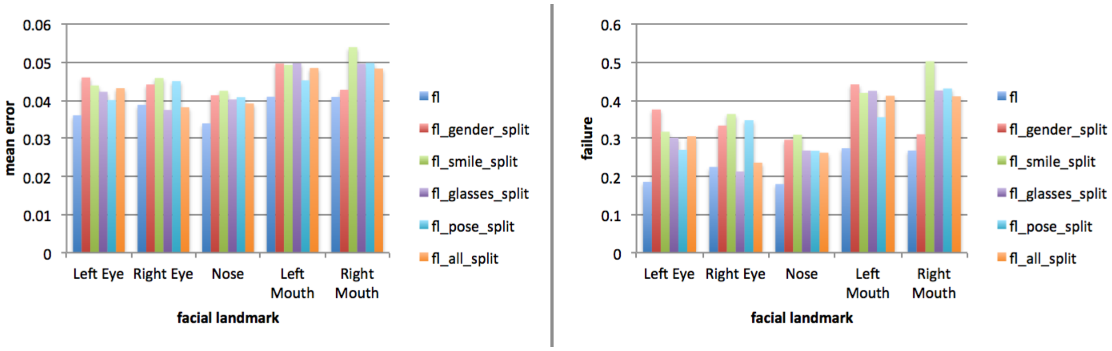


Figure 5: Facial landmark detection mean error and failure from split network

Compared to the single task facial landmark detection baseline, split networks perform worse on facial landmark detection. The split networks have higher overall mean error and failure compared to the baseline. For example, the base line failure in left eye detection is 0.186 whereas the failure in left eye detection when splitting for gender, smile, glasses, pose, and all attribute classification is 0.376, 0.317, 0.3015, 0.270, and 0.306, respectively, which are all higher than a failure of 0.186.

This drop in performance is occuring due to the different convergence rates of the two tasks. The training for the classification task would converge with less epochs than the landmark regression task, hence the classification task would overfit and harm the performance of the landmark regression task. Also, the drop in perform may also be contributed by the joint loss. Some performance on the landmark regression task could be sacrificed to improve the performance of the classification task. This may be occuring because the two tasks share the same network. Overall, some fine tuning may be necessary to migitate the affects of negative transfer between the tasks.

6.2 Cross-stitch Network vs Weight regularized Network with Cross-stitch predictions

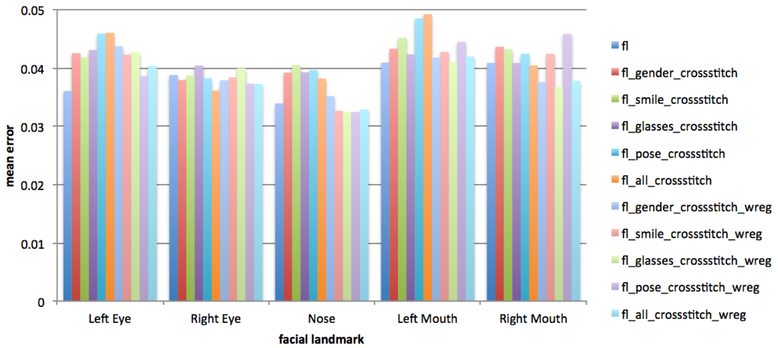


Figure 6: Mean error from Cross-stitch Network vs Weight regularized Network with Cross-stitch

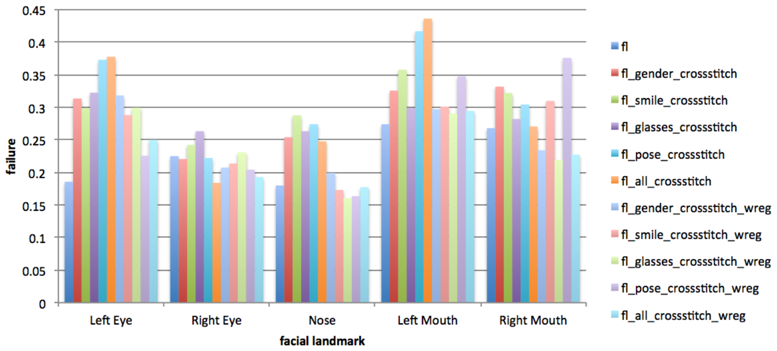


Figure 7: Failure from Cross-stitch Network vs Weight regularized Network with Cross-stitch

Depending on which facial landmark, the performance of the Weight regularized Network with Cross-stitch is similar to the single task facial landmark detection baseline on average. Though, the Weight regularized Network with Cross-stitch shows improvements over the Cross-stitch Network. In the case with all attributes used in the classification task, failure in the left eye landmark detection with the Cross-stitch Network is 0.378 whereas the failure with the Weight regularized Network with Cross-stitch is 0.250 . This shows an improvement in performance when weight regularization is introduced to the Cross-stitch Network.

Since the weight's scalar parameters a and b from equation 8 were initialized as 1 and 0, the convergence rate of attribute classification task slowed down because the weights of the two network were restrain to be similar at first. The weights differentiated in later epochs which resulted in more epochs for the landmark regression task, so the training error can be furthur reduced. Meanwhile, the landmark regression task also shares its representations with the attribute classification task more in the last layer through the cross-stitch unit to improve generalization and performance.

6.3 Weight values

In general, the cross-stitch unit weights did not change. As mentioned in section 5.2, we initialized α_S as 0.9 and α_D as 0.1. After training has been completed, most α_S and $alpha_D$ values stayed around 0.9 and 0.1, respectively. Though, the α_S and α_D in the last fully connected layer of the network for the attribute classification tasks updated to around 4 and 1, respectively. Unlike the other layers which retained $\alpha_S \approx 0.9, \alpha_D \approx 0.1$, the last layer took 80% of the original activation maps and 20% of the other network's activation maps at higher magnitudes of 4 and 1. This means that the attribute classification task has improved performance when sharing and receiving more from the landmark regression task at the last layer.

Also, the weight's scalar parameters a displayed a similar behavior. Note that the weight's scalar parameters b generally stay around 0 so it does not contribute much to the weight regularization. Though, the parameters a grew at each layer. For example, in the FL+all case, the parameters a are 0.56633, 0.712272, 0.733182, 0.718621, and 0.768055 from the first layer to the last. It can be seen that the value is growing towards 1 which means that it is desirable for the weights to be similar at the later layers. Hence, similar to the result above, attribute classification task has improved performance when it shares simiar weights as the landmark regression task at later layers.

6.4 Improvements

A few possible improvements involve fine tuning the parameters. For example, the cross-stitch units could have a higher learning rate compared to the rest of the network. This would emphasis the importance of the cross-stitch units and update the cross-stitch units more through backpropogation. Similarly, the learning rates of the scalar parameters in the weight regularization could be higher for similar reasons.

In addition to varying the learning rates, the initialized values of the cross-stitch units and weight regularization scalar parameters could be varied. Instead of enforcing some sharing between the networks with intialized values of α_S as 0.9 and α_D as 0.1, it could set α_S as 1 and α_D as 0 which means no sharing at first and allow the networks to learn how much to really share. Overall, parameter fine tuning could show improvements over the current parameters.

7 Conclusion

In this paper we have done two different types of network for facial landmark detection under the umbrella of multi-task learning. The first one was traditional split architecture where we split the network at the last layer assuming maximum representation sharing among the tasks and in the second case we applied a cross-stitch unit adopted from a recent research. This unit generalizes the multi-task learning irrespective of the degree of matching among the tasks at hand. Finally we proposed a regularization among the weights to improve the performance of cross-stitch network. It is shown that weight regularization improves the performance of cross-stitch network without any task specific fine tuning.

Acknowledgments

We would like to thank Petar Stojanov and Devendra Chaplot for suggesting relevant papers, authors and providing guidance for this topic.

References

[1] Misra, I. & Shrivastava, A. & Gupta, A. & Hebert, M. (2016) *Cross-stitch Networks for Multi-task Learning*, arXiv:1604.03539 [cs.CV]

[2] G. Obozinski & B. Taskar & M. I. Jordan. (2010) *Joint covariate selection and joint subspace selection for multiple classification problems*. Statistics and Computing, 20.

[3] R. Collobert & J. Weston. (2008) *A unified architecture for natural language processing: Deep neural networks with multitask learning*. In ICML

[4] A. Argyriou & T. Evgeniou & M. Pontil. (2008) *Convex multi-task feature learning*. JMLR, 73.

[5] S. J. Pan & Q. Yang. (2010) *A survey on transfer learning. Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359

[6] Q. Zhou & G. Wang, K. Jia & Q. Zhao. (2013) *Learning to share latent tasks for action recognition*. In ICCV.

[7] Zhanpeng, Z. & et al. (2014) *Facial landmark detection by deep multi-task learning*, European Conference on Computer Vision. Springer International Publishing

[8] R. B. Girshick. (2015) *Fast R-CNN*. In ICCV

[9] Yosinski J & Clune J, Bengio Y & Lipson H. (2014) *How transferable are features in deep neural networks?* In Advances in Neural Information Processing Systems 27 (NIPS '14), NIPS Foundation

[10] Murugesan, K. & Liu, H. & Carbonell, J.G &, Yang, Y. (2016), *Adaptive Smoothed Online Multi-Task Learning*, Neural Information Processing Systems

[11] Rozantsev, A. & Salzmann, M. & Fua, P. (2016) *Beyond Sharing Weights for Deep Domain Adaptation*, arXiv:1603.06432 [cs.CV]