Pattern Recognition and Machine Learning
Final Project
Aravind Chandradoss

1) The entire project is carried out on **TensorFlow** and the results are shown using **TensorBoard**, which logs the training data at every epoch. For this project, two datasets are mainly used. They are **CIFAR-10 and AR-face dataset**. In order to carried out the project, few assumption – such as setting **maximum epoch to 20**, restricting the convolutional and fully connected **layers** to a **maximum of 3 and 4 respectively** - were made in order to account for the computational capability.

This project is mainly carried to understand the insight of neural network. So, in most part of the report, importance is given on comparing the differences between the networks with different configurations. For this project, **nearly 170 neural networks were trained and compared.**

Before we discuss and compare, it would be more meaningful to understand the assumptions and restrictions that were taken in consideration.

Assumptions:

They are as follows,

- The **maximum epoch** was set to **20** for the training.
- The **maximum number of convolutional layer** in each network was set to **3** (with maximum **filter** count of **64** and maximum **window** size of **7x7**)
- In most of the developed network, convolutional layer is followed by max pooling layer (with 2x2 window). In some cases, in-between convolutional and maxpooling layer, additional convolutional layer are also added (in other works, **the flow is either conv→maxpool or conv→conv→maxpool**).
- The **maximum number of fully connnected layer** in each network was set to **4** (with **maximum node of 256** in some cases and **128** in some cases)
- Since the network, at maximum, is less than 10 layers, the **learning rate and momentum values were set to 0.001 and 0.9 respectively** [3, 5, 6]. These values were considered to be **constant** and were not updated as in adaptive learning.
- The training images were **preprocessed only by normalization (normalized to 0→1)**. Although, the training images could be preprocessed, But, the preprocessing is not carried out, as I heuristically expect the network to learn, by itself, from raw data.

Work carried out:

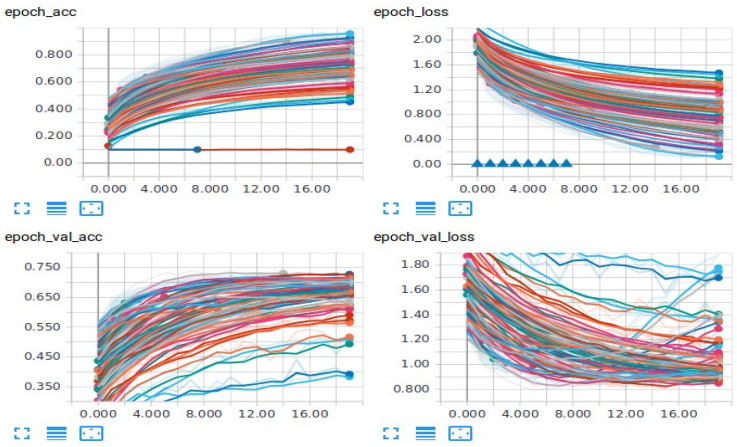
In order to compare different neural networks, I ran nearly 170 Networks in Tensorflow. The main tasks that I took are as follows,

- To infer the effect of **increasing** the number of **convolutional layer**, also with respect to increase in **convolutional filter count and window size**
- To infer the **effect** of adding **fully connected layer**, also with respect to increase in **number of nodes** in fully connected layers.
- To compare different activation function such as **relu, selu, tanh and hard sigmoid.**
- To compare different optimizer such as **adam, nadam, AdaGrad and AdaMax**
- To compare the effect of adding **dropout layer**, in particular, with respect to **dropout values of 0.2, 0.5, 0.7.**

For comparison purpose, I took the evaluation loss and the training time as the main criteria. As said earlier, all result shown in **TensorBoard** and corresponding log files are added with this report for reference.

The training curve of all the networks is shown below,

Fig 1, Training results of all CNNs



From the graph, it can inferred that, we can not trust the epoch_acc and epoch_loss as they always shows improvement. Therefor, here after **all comparison are made with respect to validation accuracy and validation loss.**

For this project, I followed the idea of **AlexNet and GoogLeNet.** The model I preferred to use is as follows,

Input → n-time(Conv→Conv→Maxpool) → few-dropout(optional) → m-time(fully_connected) → few-dropout (optional) → final softmax layer → output

Effect of Convolutional layers:

The Convolutional layer are varied by layer count, window size and filter count.

- For 1 convolution layer with 32 filter (with 0 dense layer), the networks started to overfit after 5 epoch itself.
- In next training, the convolutional layer were increased in steps with different filter count (say, 64, 32, 24).
- It can inferred from the graph that increasing the convolution can reduce the val_loss significantly, and thus the val_acc.
- It can also be inferred that the filter count also improves the classification. Convolution with 64 filters is better than 32 or 24 filters. But, increasing filters might result in slow training. (it roughly tripled the training time)
- With respect to window size, it was varied in step suzh as 1*1, 3*3, 5*5 and 7*7. Although, 7*7 showed slight better classification, the same accuracy was reached with 5x5 with one additional convolutional layer. I.e instead of using p-number of conv with 7*7, we can use p+1 conv with 5*5 or 3*3, which can provide almost a similar result with much lesser training time.

Effect of Fully connected

Dense layer are added after conv layer. The nodes in each dense layer are varied from 24 to 128 nodes. (24, 32, 64, 84, 128)

- The addition of fully connected layer (or the nodes in each layer) improved the validation accuracy appreciably (1-2%) however it did not increase the training time much, unlike convolution layer which increased the training time tremendously.

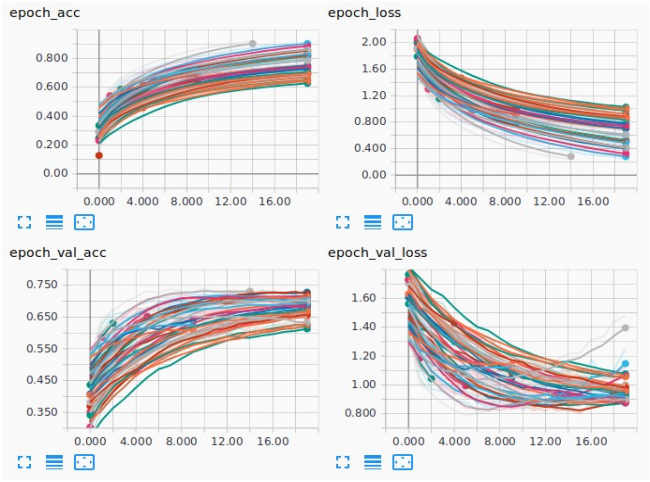


Fig2 Effect of Convolutional and Fully connected layers

Effect of activation

The activation that were used are relu, selu, tanh, and hard-sigmoid. By default, relu was used for analyzing other criteria.

- Although, selu (Bottom, U-shapped, blue curve in val_loss graph) gave faster learning, after 5-6 epoch the network started to overfit the data. Therefore, selu cannot be used throughout the training. But it can also be inferred that selu can work as appreciable activation function during the initial training. But, later it has to changed to some other activation such as Relu and tanh.
- Hard sigmoid (Top blue) shows a steady learning, but it makes the network to learn comparatively slower than Relu and Tanh.

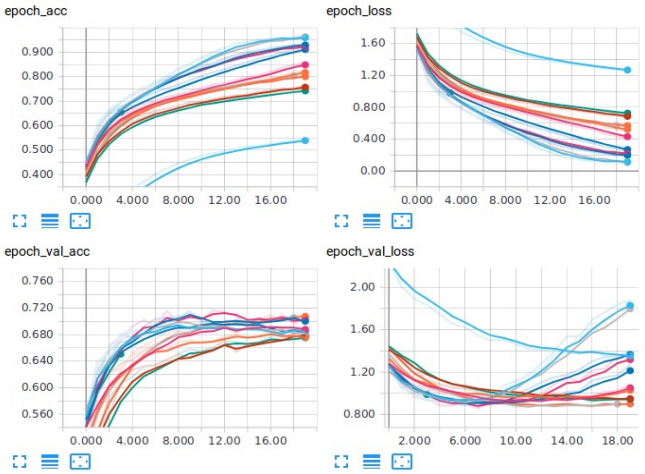


Fig 3, Effects of Activation

- Tanh gave an intermediate result. **In short, Relu is a meaningful activation, and for initial training, relu can be replaced with selu.**

Effect of Dropout

For this part, Three different dropout layer namely such as 0.2, 0.5, and 0.7, are used.

- From results, it can be inferred that 0.2 dropout aided the network to learn for generic feature. It improved the accuracy by 1%.
- Since, the network was trained only for 20 epoch, the effect of 0.5 and 0.7 was not evident. But, heuristically it can be inferred that the dropout forces the network to learn more generic feature by removing the co-adaptation between the neurons.

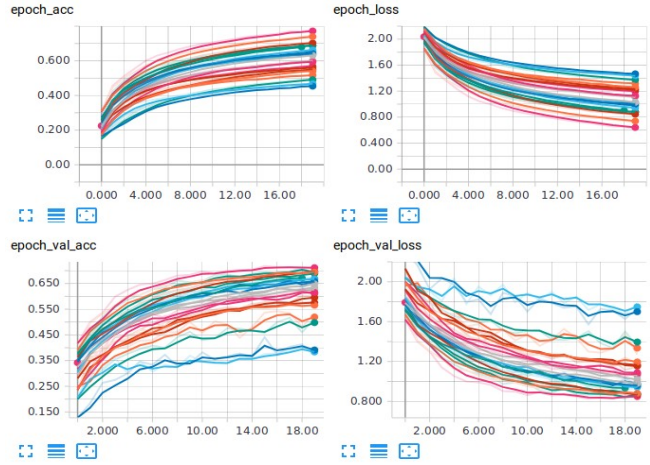


Fig 4, Effects of Dropout

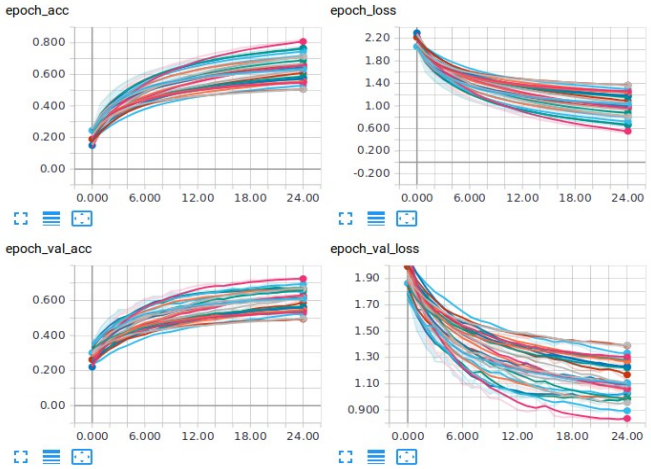
Effect of Optimizer:

The optimizer namely adam, nadam, adagrad and adamax were used on ARdataset.

- AdaGrad performed well and help in faster convergence.
- From training, the performance relations based on speed and validation loss are as follows,

AdaGrad > Nadam > AdaMax > Adam

Fig 5, Effect of Optimizers



Results and Conclusions:

The best CNNs for two datasets are as follows;

- **CIFAR:**
  2*(conv-32 filter)→dropout(0.2) →
  4*(fully_connected with 84 nodes)→dropout(0.2)→
  fully_connected (softmax layer)
  with Adam optimizer
- **AR_dataset:**
  3*(conv with 128 filters)→
  4*(Fully connected with 64 nodes)→
  Fully connected (Soft max) with Adagrad optimizer

Table 1, Accuracy of different methods:

Dataset	PCA (%)	MixerModel (%)	CNN Top-1 (%)	PreTrained GoogLeNet (%)	
CIFAR	30	~	**71-74%**	90-95	100 for top 5 guesses
AR	89	97.5	**68-72%**	90-94	100 for top 5 guesses

The Pros and Cons:

Statistical Methods (PCA, ICA, LDA, MM):

- In case of statistical models, The classifications are fully based on correlation of the input data. The statistical methods will perform poorly when the input data lags correlations (pixel correleation).
  For eg, In our case, the results for AR_database are better in case of statistical methods (PCA and MM), This is because, all the training image are faces (with same alignment and size), that is, every pixel at a specific position in a image will correspond only to specific feature (say the eye, ear or mouth). But in case of CIFAR, There is no pixel correlation, in other words, the pixels at a specific position in the images will be different and will represent different features in different image (look the CIFAR data for reference). This is the main reason for low classification accuracy for CIFAR dataset with PCA.
- Statistical Methods are more meaningful when we have infinite set of data. But in reality, we can not have infinite dataset. In case, if we have such dataset. Any statistical methods can classify images with bayes error.
- Statistical methods does not account for image deformations. That is, if a image is tilted, sheered, mirrored, cropped or any such image deformation is done, these methods might not perform well.

Neural Network (CNN, RNN):

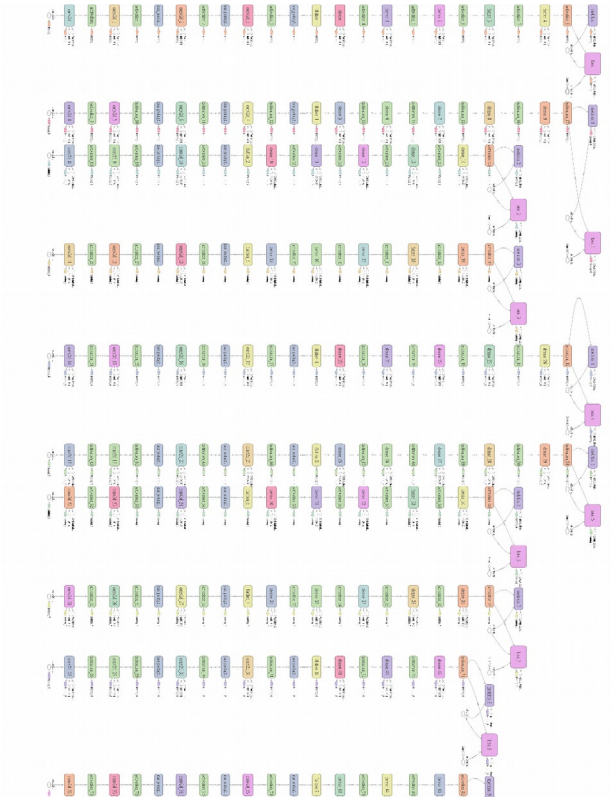
- In case of neural networks, the above mentioned problem of image deformation is highly reduced. Neural Networks can be used to classify tilted, sheered, mirrored or any such images without any issues. But the assumption is that the network is trained enough.
- The drawback of the neural networks are as follows. The procedures to design optimal architecture and to select appropriate training algorithm are still under research. Neural networks need voluminous data for training. It needs advanced hardware such as GPU, TPU (Tensor Processing Units) for the training purpose.

References:

[1] Alex K et al , ImageNet Classification with Deep Convolutional Neural Networks. Communication of the ACM, June 2017, Vol 60, pp 84-90.
[2]Szegedy Christian *et al*. Going Deeper with Convolutions. In Proceeding of the IEEE Conference on CVPV, June 2015, pp 1-9.
[3] Karen Simoyan and Andrew Zisserman, Very Deep Convolutional Neural Networks for Large Scale Image Recognition, ICLR,April 2015.
[4] Kaiming He *et al.,* Deep Residual Learning for Image Recognition, arXiv:1512.03385, Vol 1, December 2015.
[5] Gao Huang *et al.,* Densely Connected Convolutional Networks, arXiv:1608.06993, Vol 5, January 2018.
[6] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. CoRR, abs/1312.4400, 2013.
[6] https://pythonprogramming.net – Got a huge help from here!
[7] https://keras.io – Documentation was very useful here!

CIFAR

The following is just a part of the total networks. Plz, refer TensorBoard for more details



AR Face Database:

Just one CNN is shown below, refer to TensorBoard for more details.