AlexNet Experiments Results

Base Model:

1. Activation: ReLU
2. Dropout(p=0.5)
3. Pooling: Overlapping
4. Optimizer: SGD (with momentum and weight decay)

Training:

- **90 epochs** (validation accuracy measured after each epoch)
- Test Accuracy reported for model with **best validation accuracy**.

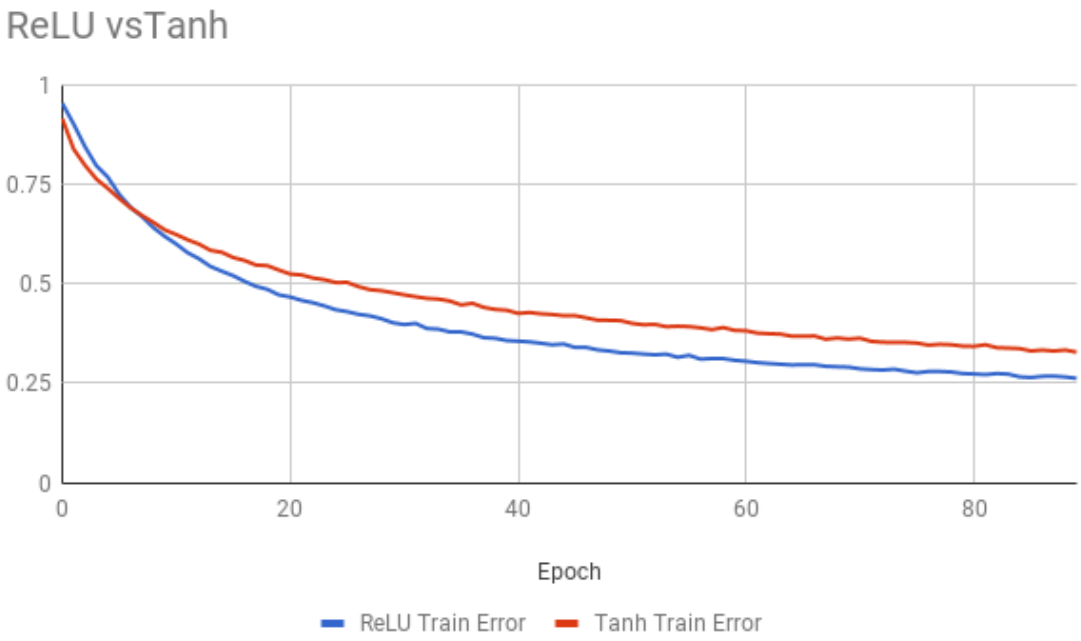
Note: The following results are w/o local response normalization. I also performed all the experiments with LRN, but they took more than twice the training time and didn't give any improvement. The results for the models with LRN are also included with the submission.

Experiment	Training Accuracy (%)	Training Time (90 epochs)	Test Accuracy (%)
Base Model	73.76	167m 1s	72.89
Base + Tanh	67.17	163m 10s	66.78
Base + No Dropout	82.15	184m 19s	69.88
Base + Non-overlapping pooling	73.07	163m 18s	71.75
Base + SGD	54.81	291m 36s	61.18
Base + SGD(with momentum)	74.58	175m 9s	72.73
Base + Adam (lr=0.001)	34.42	166m 60s	43.14

1. **Base Model**

Test Time Per Example: ~2 ms
Final Test Error: 27.11%

2. **Relu vs Tanh**



3. **Dropout v/s No Dropout**

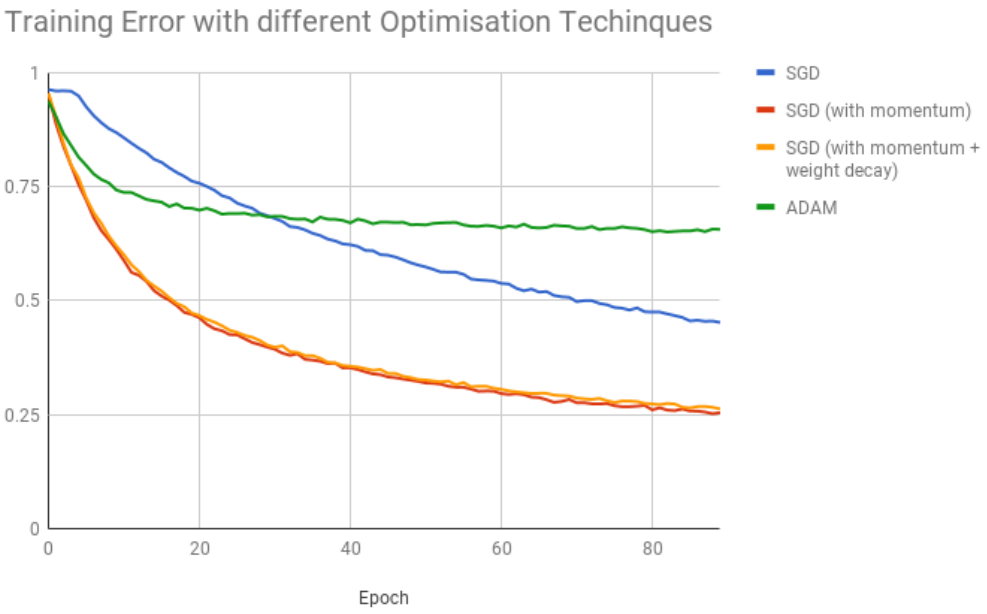
Type	Train Accuracy (%)	Best Validation Accuracy (%)	Test Accuracy (%)
w/ Dropout	73.76	74.39	72.89
w/o Dropout	82.15	71.95	69.88

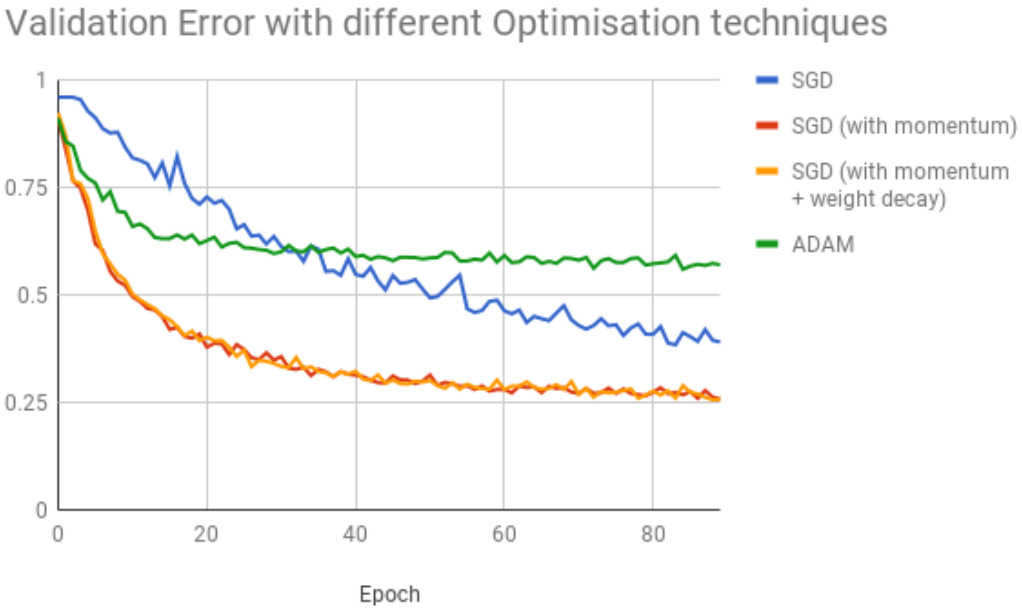
4. **Overlapping Pooling v/s Non-Overlapping Pooling**

Type	Train Accuracy (%)	Best Validation Accuracy (%)	Test Accuracy (%)
Overlapping	73.76	74.39	72.89
Non-overlapping	73.07	73.41	71.75

As can be observed, from the table, overlapping pooling clearly improves performance over non-overlapping pooling. According to the paper, it's because overlapping pooling makes it more difficult to overfit and hence improves test performance. I also found in an reddit AMA by Geoffrey Hinton, that non-overlapping pooling tends loses positional information which required to detect relationship between parts of an object and that overlapping allows some of this information to be preserved.

5. **Optimization Techniques**





6. Bonus
   In order to speed up the model training I optimized the DataLoader. This led to a per epoch time of under 2min during training. This allowed me to hand-tune the *lr* scheduling for the base model. I decreased the learning rate whenever the rate of decrease of model's validation error got low.
7. Best Model
   Validation Accuracy: 77.12%
   Test Accuracy: 76.22%
   Link: https://drive.google.com/open?id=0By07sE0zY59RSER0RGF6STNzd1k