**Assignment 4: Predicting MNIST data labels using Tensorflow**

For this assignment, I have implemented a tensorflow model to predict the MNIST data labels. The dataset contains 60,000 examples for training and 10,000 examples for testing. The digits have been size-normalized and centered in a fixed-size image (28x28 pixels) with values from 0 to 1. For simplicity, each image has been flattened and converted to a 1-D numpy array of 784 features (28\*28).

For this analysis, I am using a 3-layer CNN with max 2d pooling and tuned the hyperparameters to find the best fit model to predict the labels

|  |  |
| --- | --- |
| Activation Function | Accuracy |
| Relu | 90.06 |
| TanH | 85.7 |
| Leaky rectified linear unit (Leaky ReLU) | 87.59 |
| Crelu | 87.01 |
| Exponential linear unit (ELU) | 88.46 |
| Scaled exponential linear unit (SELU) | 88.97 |
| Softsign | 88.58 |
| SoftPlus | 10.28 |
| Softmax | 89.2 |
| Sigmoid | 98.2 |
|  |  |
| Epochs | Accuracy |
| 20 | 61.5 |
| 50 | 94.94 |
| 100 | 96.48 |
|  |  |
| Gradient Estimation | Accuracy |
| Stochastic Gradient Descent | 0.1099 |
| Adagrad | 0.1546 |
| RMSProp | 0.1413 |
| ADAM | 0.8676 |
| NAG | 0.2341 |
| Adadelta | 0.1035 |
| Momentum | 0.1145 |
|  |  |
| Network Architecture | Accuracy |
| Number of layers | 0.901 |
| Size of each layer | 0.952 |
| Connection type | 0.9456 |
| Pre-trained components? | 0.9534 |
|  | 0.9622 |
|  |  |
| Network Initialization | Accuracy |
| 0 | 94.48 |
| Uniform | 61.52 |
| Gaussian | 95.4 |
| Xavier Uniform | 96.91 |
| Xavier Gaussian | 96.28 |
|  |  |
| Loss-Function | Accuracy |
| Quadratic cost (mean-square error) | 90.32 |
| Cross-Entropy | 96.59 |
| Hinge | 89.1 |
| Kullback–Leibler divergence | 88.4 |
| Cosine Proximity | 61.3 |

Final Result:

The hyperparameters used giving best results were:

Activation Funtion: Relu

Epochs: 100

Gradient estimation: Stochastic Gradient Descent

Network Architecture: 5 layers and 512 nodes

Network Initialization: Xavier\_Gaussian

Loss function: Cross entropy