4-class problem

Optimizer – Adam

* Learning rates – 0.1, 0.01, 0.001, 0.0001
* Loss – Categorical cross entropy

Batch size – 128, 256, 512, 1024

Epochs – 10, 100, 1000, 2000

Metric – Accuracy

* Early stopping
  + Monitor – val\_loss
  + Min\_delta=0
  + Patience=2
  + Mode=’auto’

Layers

* Regularization – Dropout and L1 or L2
  + Lambda = 0.1, 0.01, 0.001
* Hidden layers
  + 1 or 2
  + 2 to double number of input neurons
  + Softmax, sigmoid, tanh, ReLU
* Output layer – Softmax

Opimizing ReLU

* Attempt 1
  + Using 1% of data (about 10000 samples)
  + 35 neurons, 768 samples per batch, 500 epochs, L2 reg, 0.001 for lambda: mean 0.297826, std 0.00347
    - Mean 0.320647, std 0.001050 for whole data set
  + Trying L1 and L2
    - Best acc is L2 where lambda is 0.05 and gives mean acc of 0.308112, std of 0.005134
    - Best acc for L1 was where lambda is 0.01 and gives mean acc of 0.300510, std of 0.001950
  + Trying neurons
    - Best acc is 45 neurons where acc is 0.309901 and std is 0.006778
  + Got acc of 0.317090 with std of 0.00208, indicating overfitted sample, so sample is too small to generalize, so need to use bigger sample
* Attempt 2
  + Found sample with better parameters since start of last attempt, 32 neurons, 768 samples per batch, 400 epochs, L2 reg, 0.001 for lambda: mean of 0.320992 and std of 0.001550 over entire test set
  + Will attempt to zero in on best number of epochs, then increase the size of the sample to tune on
    - 200 epochs gives best accuracy of 0.321233 with std of 0.001491
  + Will try using 10% of training data for tuning
    - Gives acc of 0.319259 and std of 0.00372
  + Trying neurons
    - Best acc is 75 neurons, with acc of 0.32178 and std of 0.00167
  + Changing lambda
    - Tried standard range, found for both L1 and L2 regularization accuracy was increasing as lambda was getting smaller, suggesting lambda is too larger (experiment 6 and 7)
    - To start with I tried setting lambda to 0, in case dropout alone is enough to avoid overfitting, and found accuracy increased to 0.32211 with std of 0.00399
    - Seems that dropout alone is enough with the current probability set to 0.5, will try adjusting this to see the effect, will not use regularization to avoid underfitting
  + Changing rate of dropout
    - Found 0.3 to be best, with accuracy of 0.32297, and std of 0.00326
  + Trying L1 and L2 reg without dropout
    - They seemed to not work with dropout, but it seemed a sensible idea to see what accuracies they gave when there was no dropout
    - Found best L1 to give acc of 0.32232 with std of 0.00291, found best L2 to give acc of 0.32128 and std of 0.00271
    - Decided to go with L1, as std seemed more stable
  + Changing rate of dropout
    - Tried retuning dropout to see if we could achieve a greater accuracy with that and L1 regularization too