ESE 546, FALL 2020
PROBLEM SET 4
INSTRUCTOR SOLUTIONS

Solution 1.

(c) To find the Hessian of the loss function, Let,

$$\xi = e^{-y_i w^T x_i} \tag{1}$$

$$\frac{\partial L}{\partial w_j} = \frac{1}{n} \sum_i \frac{-x_{ij} y_i \xi}{1+\xi} + \lambda w_j \tag{2}$$

Taking the derivative once again,

$$\frac{\partial L^2}{\partial w_j w_k} = \frac{1}{n} \sum_i \frac{-x_{ij}\; x_{ik}\; y_i^2 \xi}{(1+\xi)^2} + \lambda \cdot 1\{j = k\} \tag{3}$$

The Hessian is thus of the form of $X^\top AX + \lambda I$. Since $X^T AX$ is positive semi-definite by definition and λI is positive definite, therefore the best strongly convexity parameter is λ.

Let $m = \lambda$ in $\kappa = L/m$. For strong convexity, we have $\rho = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$.

Solution 2.

(a) Note that at the start of training, all the outputs have the same values (in expectation atleast). For every data point, we can expect to have $p_\theta(y|x)$ close to $\frac{1}{10}$. In other words, the neural network is approximately predicting a uniform random distribution at the start of training. This corresponds to a loss of

$$\mathcal{L} \approx \frac{1}{n} \sum_{i=1}^{n} -\log_e(\frac{1}{10}) = 2.302$$

(b) The setup was run for around 180 iterations in order to pick out the best learning rate. The learning rate started at 0.00001 with a weight decay of 0.001. Note that the architecture was changed (dropout from first layer was removed). The values of η^* was programmatically determined to be: 2.404 (this values changes slightly on different runs/initializations).

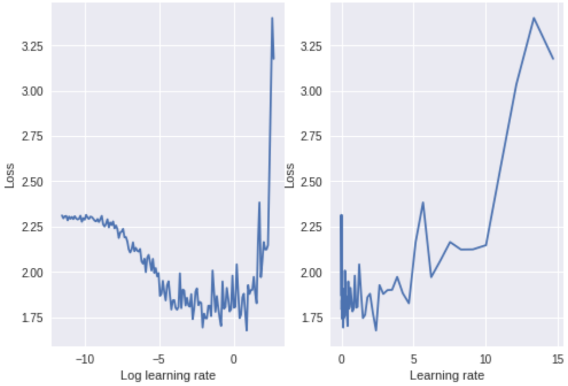

FIGURE 1. Loss against the learning rate

(c) Training and Validation cruves

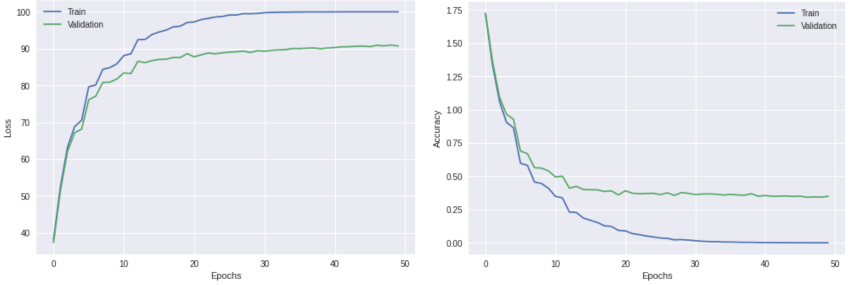

FIGURE 2. The Train and Validation accuracies (left) and the loss (right) on the entire data plotted after every epoch.

(d) The 3 scenarios were trained (this time with dropout in the first layer). The reason for adding dropout is to also show that the performance is poorer in this case. The test was not attempted without dropout (due to constraint on time). The 3 cases are:
   (a) $\eta_{\text{max}} = 0.1$, $\rho = 0.9$
   (b) $\eta_{\text{max}} = 0.5$, $\rho = 0.5$
   (c) $\eta_{\text{max}} = 0.1$, $\rho = 0.9$

   The first two cases has similar train and validation errors/losses as observed in the plots, indicating that the heuristic is accurate.

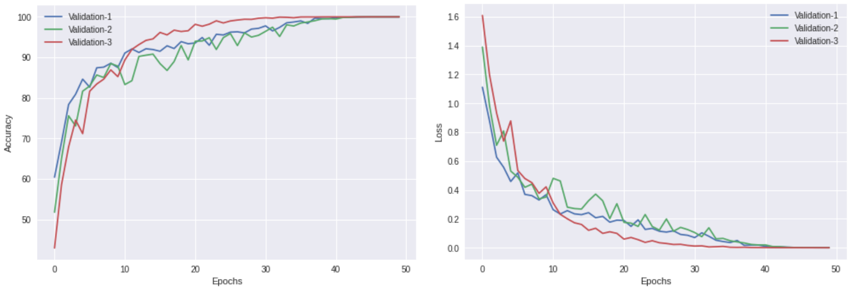

FIGURE 3. Training Accuracy (left) and Losses for 25 epochs (evaluated on entire data at the end of the epoch)