# STAT 535 Homework 5

Out Feb 25, 2026

Due March 4, 2026

©Marina Meilă and Benjamin Schneider
mmp@uwaterloo.ca

*All plots must be included in the .pdf writeup*

**Problem 1 – Decision regions for the neural network**
In this problem, the inputs are of the form $[x_1\ x_2]^T \in [-10, 10]^2$ and the dummy variable $x_0 \equiv 1$.

Consider the following two-layer neural network

$$\begin{aligned}
f(x) &= \beta_0 + \sum_{k=1}^{5} \beta_k x_k^{(1)} && (1)\\
x_k^{(1)} &= \phi(\sum_{j=0}^{2} w_{jk} x_j), \text{ for } k = 1:5 && (2)\\
x &= \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} && (3)\\
W &= [w_{jk}] \;=\; 20 \begin{bmatrix} 1 & 0 & 2 & 2 & 2 \\ 1 & 1 & 0 & -1 & -0.5 \\ -1 & 1 & -1 & 0 & 1 \end{bmatrix} && (4)\\
\phi(u) &= \frac{1}{1+e^{-u}} \text{ the sigmoid function} && (5)\\
\beta_0 &= -4.9,\ \beta_{1:5} \;=\; 1 && (6)
\end{aligned}$$

**a.** Plot the decision boundary or alternatively the decision regions $D_\pm = \{x \,|\, f(x) \lesseqgtr 0\}$ of this neural network on $[-10,\, 10]^2$.

**b.** Same as **a** for $\beta_0 = -3.9$.

**Problem 2 – Descent algorithms for training a neural network**
This problem asks you to train a neural network to classify the data sets given on the Assignments web page. The inputs are 2-dimensional, outputs are ±1, one data point/line. *Submit the code for this problem.*

Objective to minimize is $\hat{L}_{\text{logit}}(\beta, W) = -\frac{1}{n}\text{log-likelihood}(\mathcal{D}|\beta, W)$ and $\beta \in \mathbb{R}^{m+1}, W \in \mathbb{R}^{n\times m}$ are the neural net parameters. However, you will also track the misclassification error $L_{01}$ during training and at testing.

The algorithm is steepest descent with fixed step size. You need to implement the algorithm yourself.

**a.** Plot the data set $\mathcal{D}$ given `hw5-nn-train-100.dat` $\in [-10, 10]^2$ representing each class with a different color or symbol.

**b.** Based on the plot in **a.**, is it possible to get $\hat{L}_{01} = 0$ for $m = 2$? Explain in 1-2 sentences.

**c.** Implement GD for a two layer neural network in a script `hw5-main.py`. In the main function, `m` is the number of hidden units, `eta` is the learning rate for training, and `maxiter` is the maximum number of training iterations. `tol` is the threshold for the stopping criterion.

**Tip 1**: Use `./hw5-eval.sh` to check the accuracy of the evaluation and the output format.

**Tip 2**: We will evaluate your code on a different data set than the one provided, so try to make your solution general!

**d.** Use the function you implemented in **c.** to train a neural network on $\mathcal{D}$. Set the stopping criterion $1 - \frac{\hat{L}^{t+1}}{\hat{L}^{t}} \leq tol$ with $tol = 10^{-5}$. Choose a number $m \geq 5$ hidden units and an appropriate step size $\eta$. Obtain the best empirical $\hat{L}_{01}$ you can. *Note that larger m values may be easier to train.* Explain how you chose the initial points. *It's ok/recommended to plot the data and look at it or even to make a sketch of the solution you want to find. Do not show these trials, but you can describe them briefly. Show only the results from the "best" training run you obtained. The training algorithm will converge to a local optimum. It's OK to look at this local optimum and try other initial points if the found optimum is bad.*

Report: $m$, $\eta$, number steps $T$ to stopping.

Make the following plots

1. The training loss $\hat{L}^{t}_{\text{logit}}$ and the test set loss $L^{t}_{\text{logit}}$ versus $t = 1 : T$ on the same graph.
2. The training misclassification error $\hat{L}^{t}_{01}$ and the test set misclassification error $L^{t}_{01}$ versus $t = 1 : T$ on the same graph.
3. The relative change in loss $\frac{\hat{L}^{t-1}_{\text{logit}} - \hat{L}^{t}_{\text{logit}}}{\hat{L}^{t}_{\text{logit}}}$ versus $t = 1 : T$.

**e.** If you arrived here, you succeeded to write code that trains a neural network, and track some of the variables during training. This is necessary, but not sufficient. You also need to use the plots you made to evaluate if the training is completed correctly. Based on your work in **a,b,c,d** answer the following:

1. I believe that my training algorithm has converged to a (local) optimum. **YES** **NO**
2. Has your neural network overfitted the training data $\mathcal{D}$? **YES** **NO**

**f.** Plot the final decision region superimposed on the data.

*Please make clear, well-scaled, well labeled graphs.*

**g.** Repeat steps **d, e** for the larger data set `hw5-nn-train-10000.dat`. Use the same parameter initialization as in **d**. You can use `hw5-nn-test.dat` as an eval set for your model.

**h.** Plot the final decision region of the neural network from **g.** superimposed on the plot from **f.**. Okay to have a single plot for **f, g**. *Do not plot the large data set to avoid cluttering the plot.*