ASSIGNMENT I
Regression and Classification

SH2150 Machine Learning in Physics

October 2025

You are using a very fast detector to measure the energy spectrum of gamma photons emitted by an unknown radioactive nuclide. After some initial experiments you conclude that this nuclide emits two gamma rays with energies E_A (first photon) and E_B (second photon) in very rapid succession, with a time difference on the order of a picosecond. The photons are emitted in random directions, and therefore you only detect one of them in most cases. However, in some rare cases you manage to measure both and determine the order they arrive in so that you can label them A and B.

The photons interact with the detector through compton scattering and you measure the energy E and scattering angle θ for each photon. Some photons also photointeract in the detector, but their energy is outside the detector's range of validity ($E > 650$ keV) so you cannot get a trustworthy energy measurement from these, and therefore you discard them.

Your mission is to

- classify the photon events into two categories, namely A and B,
- estimate the incident photon energies E_A and E_B,
- and finally identify the nuclide.

1. The logistic regression model for classification is given by

$$\hat{y} = \sigma(\mathbf{W}\vec{x} + b) = \frac{1}{1 + e^{-(\mathbf{W}\vec{x}+b)}},$$

where we assume that the target variables $y^{(i)}$ are independently Bernoulli distributed with parameters $\hat{y}^{(i)} = \sigma(\mathbf{W}\vec{x}^{(i)} + b)$, for $i = 1, \ldots, n$. Observe that $\hat{y} \in (0, 1)$ and can therefore be interpreted as a probability.

(a) Show that fitting the model by maximizing the likelihood $L(\mathbf{W}, b \mid \vec{x}, y)$ of the data $(\vec{x}^{(i)}, y^{(i)})_{i=1}^{n}$ under the model w.r.t. $\mathbf{W}, b$, is equivalent to minimizing the cross-entropy (CE) loss defined by (2p)

$$L_{\text{CE}}(\mathbf{W}, b) = -\frac{1}{n} \sum_{i=1}^{n} y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log\left(1 - \hat{y}^{(i)}\right). \tag{1}$$

(Hint: write out the log-likelihood of the Bernoulli distribution and simplify).

(b) Derive an expression for the decision boundary when we have two predictor variables. That is, if $\vec{x} = (x_1, x_2)^\top$, for what $\vec{x}$ is the model $y = \sigma(\mathbf{W}\vec{x} + b)$ most uncertain as to which class $\vec{x}$ belongs to? (2p)

(c) Fit the logistic model to the `measuredData.mat` dataset by minimizing the CE loss, taking the scattering angle and measured energy as input variables $x = (\theta, V)$. Use the Adam optimizer with learning rate 10^{-2} and run 10 000 epochs. Using the formula found in 2b) plot the found decision boundary on top of the data, and report the found coefficients $\mathbf{W}, b$. (2p)

If you did not find a formula in 2b) for the boundary you can use the approximate formula $x_2 = -115 + 810x_1$.

(Hint: in PyTorch you can use the Adam optimizer to minimize the loss function via `torch.optim.Adam`)

2. The logistic model can be interpreted as a simple neural network that first runs the input x through an affine function $\vec{x} \mapsto \mathbf{W}\vec{x} + b$ and then spits out a single sigmoid activation $y = \sigma(\mathbf{W}\vec{x} + b) \in [0, 1]$. We can expand this into an arbitrary feed-forward neural network by adding more (hidden) layers with more nodes, and play with the choice of activation functions[1] to get the model

$$y = \sigma_l(\mathbf{W}_l \cdots \sigma_2(\mathbf{W}_2\sigma_1(\mathbf{W}_1\vec{x} + b_1) + b_2) \cdots + b_l)$$

where σ is a choice of activation function properly chosen for the case at hand.

(a) Instead of the logistics model above, fit a neural network with a two hidden layers of dimension 32 followed by 16, with tanh activations, and a single output node with sigmoid activation, using the CE loss and the Adam optimizer. Train it for 7 500 epochs with learning rate 10^{-3}. Make sure you get a reasonable fit, if not, run the training again. Plot the decision boundary on top of the data and comment on its shape in comparison to the one found in 1(c). Are there any differences? (2p)

(b) Experiment a bit with the architecture of your neural network by adjusting the number of hidden layers and the number of nodes per layer. Try to achieve the highest possible training accuracy while keeping the network as simple as possible. Is this a good classifier, how would it perform on unseen data? (1p)

3. Using your classification of the data (either classifying using the model from problem 1 or problem 2) we shall now fit one regression model to each class respectively in order to infer the incident energies E_A, E_B.

(a) Recall the Compton formula (1p)

$$E' = \frac{E_0}{1 + \frac{E_0}{m_e c^2}(1 - \cos\theta)}$$

that describes the energy E' of a photon after having Compton interacted with angle θ, where E_0 was the initial energy. Let $t = 1 - \cos\theta$ and Taylor expand the measured energy $E(t) = E_0 - E'(t)$ in terms of t.

(b) Fit each of the two classes of data to the regression model (3p)

$$y = \mathbf{W}\vec{x} + b$$

where now $\vec{x} = (t, t^2, t^3, \dots)^\top, y = E, b = 0$, using mean square error loss (`nn.MSELoss()`), with learning rate 10^3 for 25 000 epochs. Keep at least the cubic terms. Plot both fitted curves in the same plot together with the data.

(c) Derive a relationship $E_0 = E_0(W_i)$ between the found Taylor coefficients W_i and the initial energy E_0. (1p)

[1] activation functions are applied component-wise for vector outputs

(d) Use the linear term W_0 to deduce and report your estimates for E_0^A and E_0^B. Why does the linear term give a better estimate then the higher order terms here? (2p)

(e) Fitting a regression model with mean square error loss implicitly means that we assume the errors (noise) to be identically and independently distributed (iid) normal random variables. In this case, we have assumed the noise to be approximately iid normal, but looking closely at the data we can see that this is not the case. Describe a way to handle the varying noise to get a more robust fit, and investigate how the result changes when you apply this method. (3p)

4. Find a reasonable candidate for the unknown nuclide using your estimated energies. For this question *only*, you are allowed to ask an AI chatbot for help! (1p)

Final question: Did you use an AI tool (other than the machine learning models you trained in this exercise) for anything else than information searching, when solving this problem set (including problem 4)? If so, please write a **brief statement of how you used AI.**.

Total number of points: 20

Remember to motivate your answers wherever applicable.

Good luck!