Problem 2: Neural Networks

Claudio López, Diogo Silva

Universidad de Chile

October 21, 2024

Problem 2 overview

Goals

- Build a neural network capable of diagnosing breast cancer (binary classification)
- Define a loss function
- Find the optimal network parameters: activation function, optimizer, number of layers and number of neurons
- Investigate the effect of normalizing the training data
- Build a classifier for a multiple class problem

Exercise 1: Introducing Log-Likelihood

Maximum Likelihood Estimation

$$\max_{\theta} \prod_{i=1}^{N} p_{\theta}(y^{(i)}|x^{(i)})$$

where for each θ,

$$p_{\theta}(y|x) \geq 0,$$

$$\sum_{i=1}^{N} p_{\theta}(y^{(i)}|x^{(i)}) = 1$$

Exercise 1: Introducing Log-Likelihood

Finding optimal θ

$$
\begin{aligned}
\hat{\theta} &= \arg\max_{\theta} \prod_{i=1}^{N} p_{\theta}(y^{(i)}|x^{(i)}) \\
&= \arg\max_{\theta} \frac{1}{N} \sum_{i=1}^{N} \log\left(p_{\theta}(y^{(i)}|x^{(i)})\right)
\end{aligned}
$$

Exercise 2: Defining the Loss Function

Cross-entropy

Given two discrete probability p and q with the same support χ, the **cross-entropy** of p and q is given by

$$H(p, q) = -\sum_{x \in \chi} p(x) \log(q(x))$$

Cross-entropy loss function

The **cross-entropy** loss function is defined as

$$L(q, p) = \frac{1}{N} \sum_{i=1}^{N} H(p_i, q_i)$$

Exercise 2: Defining the Loss Function

Finding optimal θ

$$\begin{aligned}\arg\min_{\theta} L(p_\theta, p) &= \arg\min_{\theta} \frac{1}{N}\sum_{i=1}^{N} H(p_i, p_{\theta_i}) \\ &= \arg\max_{\theta} \frac{1}{N}\sum_{i=1}^{N} \log(p_\theta(y^{(i)}|x^{(i)})) = \hat{\theta}\end{aligned}$$

Metrics for the Classifier

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Metrics for the Classifier

Precision

Precision is the ratio of true positive predictions to the total positive predictions made by the model. It is defined as:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall

Recall measures the ratio of true positive predictions to the total actual positives. It is defined as:

$$\text{Recall} = \frac{TP}{TP + FN}$$

Metrics for the Classifier

Matthews Correlation Coefficient (MCC)

Matthews Correlation Coefficient is a more informative metric that takes into account all four categories of the confusion matrix. It is defined as:

$$\mathrm{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

MCC returns a value between -1 and 1, where 1 indicates a perfect prediction, 0 indicates no better than random prediction, and -1 indicates total disagreement between prediction and observation.

Exercise 3: Testing the first Neural Network

Network 1 Parameters

- Number of Layers: 2
- Number of Neurons: 16
- Activation Function: Sigmoid $:= \frac{1}{1+e^{-x}}$
- Optimizer: SGD

Exercise 3: Testing the First Neural Network

Average Network 1 Performance

- **Precision:** 0.6833
- **Recall:** 0.9938
- **MCC:** 0.1733

Average Network 1 Confusion Matrix

	Predicted Positive	**Predicted Negative**
Actual Positive	71.56	0.44
Actual Negative	35.16	6.84

Exercise 4: Testing Different Activation Functions

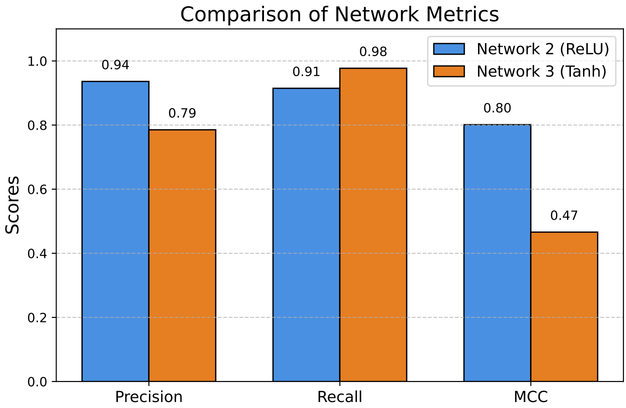
Network 2 Parameters

- Number of Layers: 2
- Number of Neurons: 16
- Activation Function: ReLU $:= \max(0, x)$
- Optimizer: SGD

Network 3 Parameters

- Number of Layers: 2
- Number of Neurons: 16
- Activation Function: Tanh $:= \frac{e^x - e^{-x}}{e^x + e^{-x}}$
- Optimizer: SGD

Exercise 4: Testing Different Activation Functions



Exercise 4: Testing Different Optimizers

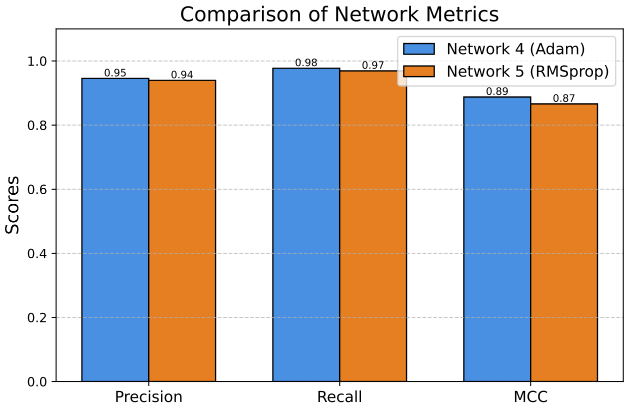
Network 4 Parameters

- Number of Layers: 2
- Number of Neurons: 16
- Activation Function: Sigmoid
- Optimizer: Adam

Network 5 Parameters

- Number of Layers: 2
- Number of Neurons: 16
- Activation Function: Sigmoid
- Optimizer: RMSprop

Exercise 4: Testing Different Optimizers



Exercise 5: Different Number of Layers and Neurons

Network 6 Parameters

- Number of Layers: 2
- Number of Neurons: 64
- Activation Function: ReLU
- Optimizer: Adam

Network 7 Parameters

- Number of Layers: 8
- Number of Neurons: 16

Network 8 Parameters

- Number of Layers: 4
- Number of Neurons: 32

Exercise 5: Different Number of Layers and Neurons

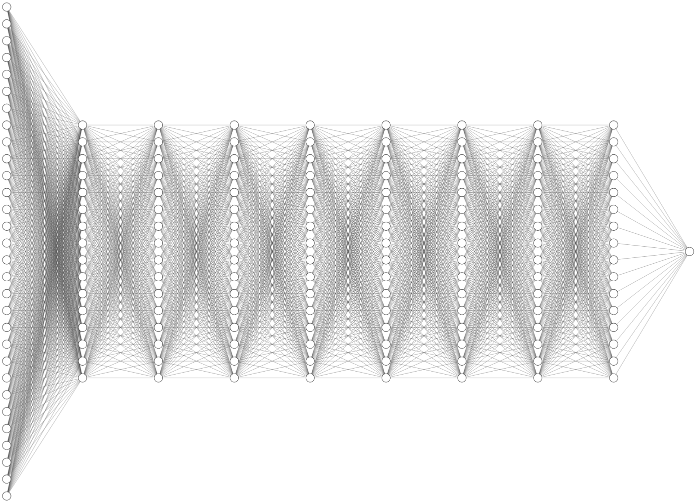


Figure: Network 7: 8 Layers, 16 Neurons

Exercise 5: Different Number of Layers and Neurons

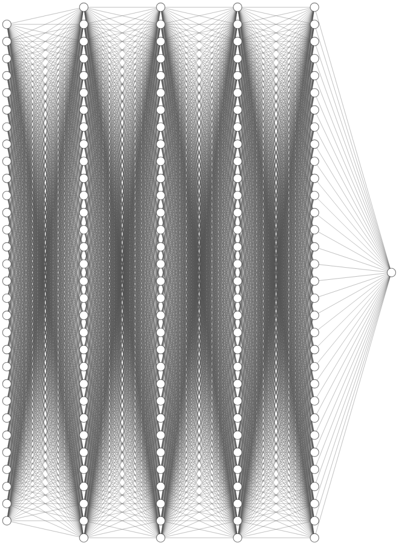
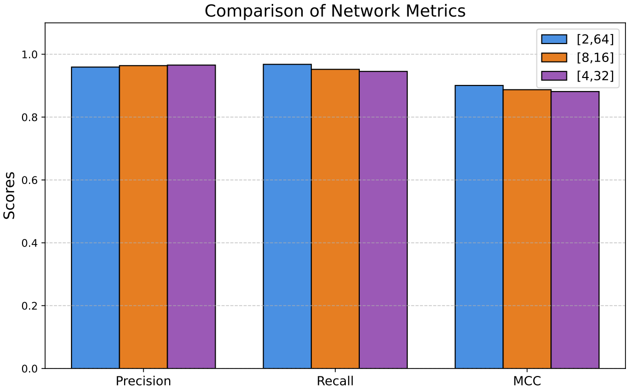


Figure: Network 8: 4 Layers, 32 Neurons

Exercise 5: Different Number of Layers and Neurons

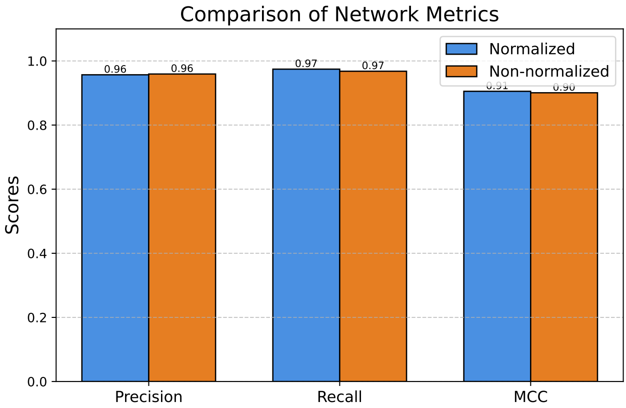


Exercise 6: Normalizing the Data

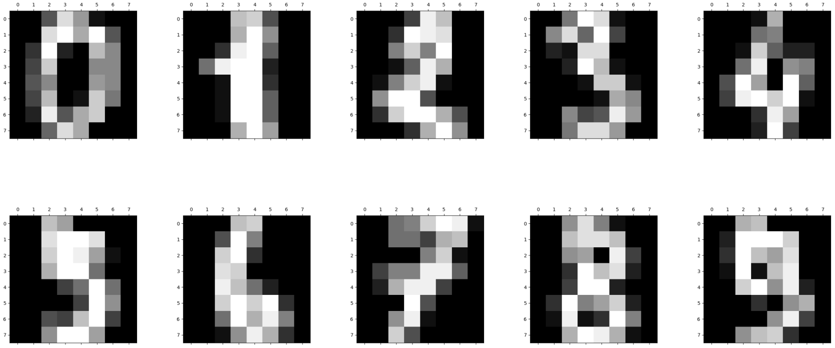
Network 9 Parameters

- Training and Testing sets normalized
- Number of Layers: 2
- Number of Neurons: 64
- Activation Function: ReLU
- Optimizer: Adam

Exercise 6: Normalizing the Data



Exercise 8: Multiple Classes Classifying



Exercise 8: Multiple Classes Classifying

Network Parameters

- Training and Testing sets normalized
- Number of Layers: 3
- Number of Neurons: 128
- Activation Function: Tanh
- Optimizer: Adam

Average Network Performance

- **Precision:** 0.9108
- **Recall:** 0.9091

Prediction Animation

External Link

Exercise 8: Multiple Classes Classifying