

CS3317 Artificial Intelligence

Coursework 3: Multi-Layer Perceptron & Backpropagation

[Example Report — For Reference Only]

Name: Zhang San Student ID: 522030XXXXX

May 6, 2026

Note: This is an example report showing the expected structure and depth of analysis. All numerical results shown as `--` are placeholders—replace them with your own experimental results. Do not copy the analysis verbatim; write your own observations based on your experiments.

1 Implementation Overview

Layer Implementations

Linear Layer. The forward pass computes $\mathbf{z} = \mathbf{X}\mathbf{W} + \mathbf{b}$, caching the input $\mathbf{X}$ for the backward pass. The backward pass computes three quantities:

$$\frac{\partial L}{\partial \mathbf{W}} = \mathbf{X}_{\text{cache}}^{\top} \cdot \nabla_{\mathbf{z}} L \quad (1)$$

$$\frac{\partial L}{\partial \mathbf{b}} = \sum_i \nabla_{\mathbf{z}_i} L \quad (2)$$

$$\nabla_{\mathbf{X}} L = \nabla_{\mathbf{z}} L \cdot \mathbf{W}^{\top} \quad (3)$$

[Explain each equation. Note that the first two are identical to CW2's gradient computation. The third is new—it propagates the gradient to the previous layer, enabling multi-layer backpropagation.]

ReLU Activation. *[State the forward formula and explain the backward pass: gradient passes through where input was positive, zeroed otherwise.]*

Sigmoid Activation. *[State the forward formula and the derivative $\sigma'(x) = \sigma(x)(1 - \sigma(x))$. Mention any caching strategy.]*

Connection to CW2

[This is an important section. Explain how your CW3 implementation builds on CW2:]

- **CrossEntropyLoss:** *[How did you reuse your CW2 softmax and cross-entropy code?]*
- **Linear.backward():** *[Explain the key insight: CW2's optimizer was secretly performing the backward pass of a single linear layer. How does the generalization work?]*
- **SGD with momentum:** *[How does this extend CW2's gradient descent? What happens when momentum = 0?]*

MLP Assembly

[Describe how layers are chained: forward pass processes layers in order, backward pass reverses order. Mention whether gradient checks passed.]

2 Hyperparameter Exploration

2.1 Hidden Layer Dimension

[Vary the hidden layer width (e.g., several values from small to large) while keeping other hyperparameters fixed.]

Table 1: Effect of hidden layer dimension on performance.

Hidden Dim	Train Acc	Val Acc	Test Acc	#Params
---	---.-%	---.-%	---.-%	-----
---	---.-%	---.-%	---.-%	-----
---	---.-%	---.-%	---.-%	-----
---	---.-%	---.-%	---.-%	-----

[Insert your sweep_hidden_dim.png here]

Figure 1: Test accuracy vs. hidden layer dimension.

Analysis: [Discuss the trend. Is there a point of diminishing returns? Do you observe overfitting at larger widths (train accuracy much higher than validation)?]

2.2 Number of Layers

[Fix the hidden dimension and vary the depth.]

Table 2: Effect of network depth on performance.

Num Layers	Architecture	Val Acc	Test Acc	#Params
---	-----	---.-%	---.-%	-----
---	-----	---.-%	---.-%	-----
---	-----	---.-%	---.-%	-----

Analysis: [Does deeper always mean better? Discuss vanishing gradients, overfitting, and convergence speed at different depths.]

2.3 Learning Rate

[Test several learning rates spanning a wide range.]

Analysis: [What happens with too-high learning rates (oscillation, divergence)? What about too-low rates (slow convergence, underfitting)? How does momentum interact with learning rate?]

Table 3: Effect of learning rate on performance.

Learning Rate	Val Acc	Test Acc	Notes
-----	---.-%	---.-%	-----
-----	---.-%	---.-%	-----
-----	---.-%	---.-%	-----

[Insert your sweep_learning_rate.png here]

Figure 2: Training loss curves for different learning rates.

2.4 Additional Experiments (Optional)

[You may also explore: activation function (ReLU vs. Sigmoid), momentum values, batch size, etc. For each, provide a table and brief analysis.]

3 Comparison with CW2

Table 4: Comparison between logistic regression (CW2) and MLP (CW3).

Method	Test Accuracy	#Parameters
Logistic Regression (CW2)	---.-%	-----
MLP — default config (CW3)	---.-%	-----
MLP — best config (CW3)	---.-%	-----

[Discuss the following points based on your results:]

- *[Why does the MLP outperform logistic regression? What is the role of non-linear activation functions in creating non-linear decision boundaries?]*
- *[How does hierarchical feature extraction in hidden layers differ from a single linear transformation?]*
- *[Despite the improvement, what limitations remain when operating on flattened pixel vectors?]*

4 Conclusion

[Summarize 2–4 key findings. For example:]

1. *[How does backpropagation generalize CW2’s single-layer gradient computation to multiple layers?]*

2. *[What architecture and hyperparameter choices gave the best accuracy–complexity tradeoff?]*
3. *[How does the MLP compare to CW2, and what limitations remain that motivate CNNs?]*