Deep learning - Homework 1

Luka Šveigl (63200301)

1 Introduction

This report outlines the implementation of a basic neural network, enhanced with features like the Adam optimizer, L2 regularization, and a learning rate schedule. Through a series of experiments, we assess the efficacy of our neural network implementation and discuss and analyze the results. During discussion, we place most importance on answering the following questions: how do different optimizers affect performance, how L2 regularization impacts performance and lastly, how the learning rate schedule affects performance.

2 Methodology

In this section, we provide short overviews of the techniques used in the implementation of the neural network.

2.1 Network implementation

Our implementation of the neural network is based on the template kindly provided as part of the supplementary material of this homework. The activation function for all layers but the last is the sigmoid function, defined by the equation

$$\sigma(z) = \frac{1}{1 + e^{-z}},$$

while the final layer uses the softmax activation function, defined by the equation

$$f(z_i) = \frac{e^{z_i}}{\Sigma_j^C e^{z_{i,j}}}.$$

The loss function used is cross entropy.

2.2 L2 regularization

L2 regularization, or weight decay, combats overfitting by adding a penalty term to the loss function. The term is proportional to the square of the weights as represented by the following equation:

$$C = C_0 + \frac{\lambda}{2n} * \Sigma_w w^2.$$

2.3 Optimizers

To implement our neural network, we implemented 2 optimization algorithms for our neural network: Stochastic Gradient Descent (SGD) and Adaptive Moment Estimation (ADAM). SGD updates the model's parameters by computing the gradients of the loss function, before adjusting the parameters in the opposite direction of the gradient with respect to the learning rate:

$$w_{t+1} = w_t - \lambda \nabla J(w_t; x_i, y_i).$$

On the other hand the ADAM optimizer is an adaptive learning rate optimization algorithm for training deep learning models. It computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1) \nabla J(w_{t-1}; x_i, y_i), \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2)(\nabla J(w_{t-1}; x_i, y_i))^2, \\ \hat{m}_t &= \frac{m_t}{1 - \beta_1^t}, \hat{v}_t = \frac{v_t}{1 - \beta_2^t}, w_{t+1} = w_t - \frac{\gamma}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t \end{aligned}$$

The parameters for our implementation of the ADAM optimzier are as follows: $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$.

2.4 Learning rate decay

Learning rate decay is a technique where the network starts of with a high learning rate and gradually decreases it over epochs. This strategy helps escape spurious local minima, prevents overfitting to noisy data, and generally refines the solution.

In our implementation of the neural network, the learning rate (η) decreases exponentially with each epoch according to the formula

$$\eta_t = \eta \times e^{-k \times t}.$$

3 Results

To evaluate our neural network, we tested it using 10 experiments, the results of which are present in table 1. Throughout all the experiments the number of epochs and batch size remained consistent, 20 and 64 respectively. In experiments that include L2 regularization, the λ parameter was consistently set to 0.01, and in experiments that include learning rate decay, the k parameter was consistently set to 0.01. All of the experiments were conducted on the CIFAR-10 data set, which represents a 10 class prediction problem.

Experiment	Schedule	L2 reg.	Optimizer	LR	Arch.	Cls. Acc.
Exp. 1	None	No	SGD	0.01	100, 100, 10	45.6%
Exp. 2	None	No	Adam	0.01	100, 100, 10	46.8%
Exp. 3	None	Yes	SGD	0.01	100, 100, 10	45.9%
Exp. 4	None	Yes	Adam	0.01	100, 100, 10	47.3%
Exp. 5	Yes	No	SGD	0.01	100, 100, 10	30.5%
Exp. 6	Yes	No	Adam	0.01	100, 100, 10	47.7%
Exp. 7	Yes	Yes	SGD	0.01	100, 100, 10	30.5%
Exp. 8	Yes	Yes	Adam	0.01	100, 100, 10	47.5%
Exp. 9	Yes	Yes	Adam	0.01	100, 256, 10	48.5%
Exp. 10	Yes	Yes	Adam	0.01	100, 256, 256, 10	48.3%

Table 1: The experiments and their results

4 Discussion

Our task was to implement and evaluate our neural network, and in doing so evaluate three scenarios: training the neural network with different optimizers (Adam and SGD), training the neural network with and without L2 regularization and training the neural network with and without learning rate schedule.

To compare the performance of the neural networks with the Adam and SGD optimizers we focus on experiments 1 and 2, as they represent the neural network without any additional improvements. We can observe that the Adam optimizer improves classification accuracy compared to SGD.

When discussing the second question, i.e. what is the difference in performance with and without L2 regularization, we focus on experiments 1 and 3 (SGD) and 2 and 4 (Adam). In both cases we can observe that L2 regularization marginally improved classification accuracy.

When discussing the impacts of learning rate schedule on classification accuracy, we focus on experiments 1 and 5 (SGD) and 2 and 6 (Adam). Interestingly, we can observe that our implementation of learning rate decay massively decreases classification accuracy of a neural network that implements SGD, while improving the classification accuracy of a neural network that implements the Adam optimizer.

Experiments 7-10 represent our attempts to increase the classification accuracy of the network as much as possible. We can observe that the biggest increase in classification accuracy comes from changing the architecture of our network, as noted in experiment 9, which boasts the best results with a three layer architecture, where the layers hold 100, 256 and 10 neurons respectively.

5 Conclusion

Our examination of different optimization techniques, regularization methods, and learning rate schedules reveals important insights into neural network performance. Overall, the Adam optimizer consistently outperformed SGD, while L2 regularization had a minor impact on accuracy.

Interestingly, the implementation of a learning rate schedule produced mixed results, decreasing accuracy with SGD but increasing it with Adam. Notably, adjusting the network architecture, as seen in Experiment 9, led to notable improvements in accuracy.

In conclusion, our study emphasizes the significance of parameter tuning and architectural adjustments in optimizing neural network performance. Further refinement may enable surpassing the 50% accuracy threshold, highlighting the iterative process of network development.