Assignment 1

Convolutional Neural Networks for Image Classification

Exercise for the course Deep Learning in Computer Vision and Remote Sensing

Summer Term 2024

Submitted By: Group 4

Members:

i. Sudur Bhattarai (uhkmj@student.kit.edu)
ii. Adersh Kolappurath Remesan (uwuxd@student.kit.edu)
iii. Amal Kareem Abdul Salam (uzrxr@student.kit.edu)
iv. Aravind Santhosh (ubtyp@student.kit.edu)
v. Nifina Joseph Vadakkoot (uqzmt@student.kit.edu)
vi. Sharath Sunil Ravi (uzgni@student.kit.edu)
vii. Tony Tom Joseph (untjw@student.kit.edu)

4 Questions (10 P)

Answer the following questions.

1. **What is the difference between an iteration and an epoch in this training scenario? (0.5 P)**

Ans.

Epoch:

- In this training scenario, Epoch is one forward and backward pass of all training samples. That means there is one complete pass through the entire training dataset in one epoch.
- During one epoch, the model sees and processes all the training samples exactly once.
- Given 4200 training samples, one epoch would involve 4200 samples/(32 samples/batch) = 131.25 iterations. Since you can't have a fraction of an iteration, this means 132 iterations (with the last batch likely being smaller than 32 samples.

Iteration:

- Since we are not processing our entire training data in one forward pass but passing number of batches and performing one pass over each pass which is then optimised through back propagation. So the number of forward pass that a network has to make to complete one epoch will be the iterations.
- An iteration refers to a single update of the model's parameters. In the context of batch training, one iteration corresponds to processing one batch of data and updating the model's weights based on the error calculated from that batch.

- With a batch size of 32, each iteration will involve processing 32 training samples in a batch.

(*Difference Between a Batch and an Epoch in a Neural Network - MachineLearningMastery.Com*, n.d.; *Epoch vs Iteration When Training Neural Networks? - Intellipaat Community*, n.d.; *Machine Learning - Epoch vs Iteration When Training Neural Networks - Stack Overflow*, n.d.; *What Is the Difference Between "Epoch" and "Iteration" in Training Neural Networks - GeeksforGeeks*, n.d.)

2. **What do the values in the last (linear) layer tell you? How are they called? (0.5 P)**

Ans. The values in the Last Linear layer **(***self.fc2 = nn.Linear(256, 2))* represent the final outcome or the prediction of the model. These values are typically called the class scores or logits.

These values convey the following information:

- Each logit represents the neural network's score for a particular class. In binary classification like in this exercise where we need to distinguish "not-Car" and "Car", there will be two logits —One value for "not-Car" class and other for "Car" class.
- The logits provide a measure of the network's confidence in its predictions. However, a Softmax function is applied to convert logits into probabilities, whereby the class with the highest probability is typically chosen as the predicted class.

(*Convolutional Neural Networks (CNNs) and Layer Types - PyImageSearch*, n.d.; *CS231n Convolutional Neural Networks for Visual Recognition*, n.d.)

3. **Which function do you need to apply in order to receive an actual classification result? (0.5 P)**

Ans. The function that is needed to apply in order to receive an actual classification result is

'torch.max()'.

_, predicted = torch.max(outputs.data, 1): torch.max(output.data, 1) finds the maximum value along the dimension 1 (class dimension) of the output logits. It returns two tensors - the maximum values and their indices (*Question about Use of Torch.Max Function to Calculate Accuracy - Vision - PyTorch Forums*, n.d.).

Maximum values are discarded by using '_,' and only keep the indices predicted, which correspond to the predicted class labels. So, 'predicted' will contain the class indices (0 for 'not-car' and 1 for 'car') for each image in the batch. These indices are mapped back to the actual class names to get the actual classification result.

4. **Why is Cross Entropy an adequate loss function for classification? Could you use an L2-Loss? Why/why not? (0.5 P)**

Ans. Cross Entropy is an adequate loss function for classification due to the following reasons:

i. The output of CNN is passed through a softmax function to obtain a probability distribution over the classes. Cross-entropy loss measures the performance of a classification model by comparing the predicted probability distribution with the true distribution (one-hot encoded labels).

ii. Cross-entropy loss can seamlessly handle binary (car vs. not-car) or multi-class classification problems by computing the loss for each class and summing them up.

iii. It penalizes confident misclassifications more heavily than uncertain ones, encouraging the model to be more discriminative.

(*Cross Entropy Loss: Intro, Applications, Code*, n.d.)

Using cross-entropy loss for this classification task is appropriate because it directly optimizes the CNN to output accurate class probabilities, which is the desired objective.

2nd Part:

Since L2 loss (e.g. mean squared error) is commonly used for regression problems, it is generally not recommended for classification tasks like car vs. not-car image classification because of the following reasons:

i. L2 loss assumes that the target values are unbounded, but in classification, the targets are one-hot encoded (0 or 1), leading to potential instability during training.

ii. L2 loss treats all misclassifications equally, regardless of their confidence or the number of classes involved.

(*Cross Entropy Loss: Intro, Applications, Code*, n.d.)

5. In Pytorch, a Softmax-Function is integrated into the Cross Entropy loss. What is a Softmax Function and what is its effect on the output? (0.5 P)

Ans. In PyTorch, the softmax function is integrated into the cross-entropy loss function for multi-class classification tasks (*How to Implement Softmax and Cross-Entropy in Python and PyTorch - GeeksforGeeks*, n.d.). The softmax function is a mathematical operation that takes an N-dimensional vector of real numbers and transforms it into a vector of real numbers in the range (0,1) which adds up to 1**.** The softmax formula is represented as:

$$pi = \frac{e^{a_i}}{\sum_{k=1}^{N} e_k^a} \quad (1.1)$$

(*Softmax and Cross Entropy Loss*, n.d.)

2nd Part:

The Softmax function transforms the raw output vectors (scores) from the final layer of a neural network into the probabilities vector, which can be achieved in the following steps:

i. **Exponential Scaling:** The exponent of each entries in the raw vector from the output layer of the neural network are calculated, whereby the higher scores become more prominent after exponentiation, while the lower scores are further minimized.

ii. **Normalization:** The exponent of each entry is divided by the sum of the exponent of all entries, this normalizes the exponentiated values into the probabilities.

iii. **Probabilities Calculation:** The values after the normalization represent the probability of each class; these are arranged into a vector representing the final Softmax output.

(*Decoding Softmax: Understanding Its Functions and Impact in AI - Zilliz Blog*, n.d.)

For example, in an image classification task, the Softmax function helps in determining the likelihood of an image belonging to each possible class, such as "car" and "not-car" in the context of this exercise.

6. Explain briefly and precise what leads to model overfitting. (0.5 P)

Ans. Overfitting is a common challenge in machine learning where a model learns the training data too well, including its noise and outliers, making it perform poorly on unseen data. The following factors lead to the model overfitting:

i. **High model complexity:** Models with many parameters, like deep neural networks, can become too flexible and fit the training data too closely, including the noise (What Is Overfitting in Deep Learning [+10 Ways to Avoid It], n.d.).

ii. **Insufficient data:** If there's not enough data, the model might find patterns that don't really exist (*What Is Overfitting? | DataCamp*, n.d.).

iii. **Noisy data:** If the training data contains errors or random fluctuations, an overfitted model will treat these as patterns (*What Is Overfitting? | DataCamp*, n.d.).

iv. **Long training:** Training the model for too long on the same data can cause it to memorize the noise and outliers instead of generalizing the underlying patterns (*What Is Overfitting? | IBM*, n.d.).

In summary, overfitting happens when a model captures the random noise instead of the underlying relationship in the training data, limiting its ability to generalize well to new, unseen data (*Overfitting in Machine Learning: What It Is and How to Prevent It*, n.d.; *What Is Overfitting? - Overfitting in Machine Learning Explained - AWS*, n.d.).

7. How would an overfitting of the classifier affect the classification metrics?

Ans. An overfitted classifier will have poor performance on unseen independent test data, which will be reflected in the following classification metrics:

i. **Accuracy:** The overall accuracy on the test dataset will be low, as the overfitted model has learned the noise and outliers in the training data instead of the true underlying patterns (*Avoid Overfitting & Imbalanced Data with Automated Machine Learning - Azure Machine Learning | Microsoft Learn*, n.d.; *Overfitting in Machine Learning and Computer Vision*, n.d.). This makes it unable to generalize well to new, unseen independent data.

ii. **Precision and Recall:** Precision (the fraction of positive predictions that are truly positive) and recall (the fraction of actual positives that are correctly predicted) will likely be imbalanced and suboptimal (*What Is Overfitting? | IBM*, n.d.). An overfitted model may have very high precision by being too conservative in making positive predictions, or very high recall by overgeneralizing and making too many positive predictions.

iii. **F1 Score:** The F1 score, which combines precision and recall, will be low due to the imbalance between these two metrics caused by overfitting (*What Is Overfitting? | IBM*, n.d.).

iv. **ROC AUC:** The area under the receiver operating characteristic curve (ROC AUC), which measures the ability to distinguish between classes, will be lower for an overfitted model compared to a well-generalized one (*Avoid Overfitting & Imbalanced Data with Automated Machine Learning - Azure Machine Learning | Microsoft Learn*, n.d.).

8. Which model performed better in your tests? Why did it perform better? Explain in detail. (5 P)

Ans. Given that we have 945 ‘not-car’ test images and only 400 ‘car’ test images, correctly classifying the majority class (not-car) should be the primary focus in the context of this exercise. So, ResNet18 model can be considered the better performer for this specific exercise. ResNet18 model performs better than the other custom CNN model due to the following reasons:

i. **Precision for Not-Car Class:** The ResNet18 model has a higher precision of 0.95 for the ‘not-car’ class, compared to 0.93 for the same class in the custom CNN model. This means ResNet18 is better at correctly identifying not-car images when it predicts the "not-car" class.

ii. **Overall Accuracy:** The overall accuracy in the case of the ResNet18 model is 0.9413075780089153, while the overall accuracy with the custom CNN model is 0.9398216939078752. So the overall accuracy with the ResNet18 Model is slightly greater than the CNN model.

iii. **True Positive Rate/ Sensitivity/ Recall**: The proportion of actual positives that are correctly identified by the ResNet18 Model is higher (0.885) than that with Custom CNN model (0.832).

iv. **AUC-ROC:** The AUC-ROC (Area Under the Receiver Operating Characteristic Curve) in case of custom CNN model is 0.91 while that with ResNet18 Model is 0.93.

In summary, the ResNet18 model can be considered the better performer for this specific exercise, where correctly classifying the majority class (‘not-car’) is the primary objective due to the class imbalance in the test data. Nevertheless, the custom CNN model performed better on the minority class (‘car’). The ResNet18 model's better performance on the majority class (‘not-car’) can be attributed to its deeper architecture and the use of residual connections, which may have helped it learn more discriminative features for the ‘not-car’ class.

References

Avoid overfitting & imbalanced data with Automated machine learning - Azure Machine Learning | Microsoft Learn. (n.d.). Retrieved June 8, 2024, from https://learn.microsoft.com/en-us/azure/machine-learning/concept-manage-ml-pitfalls?view=azureml-api-2

Convolutional Neural Networks (CNNs) and Layer Types - PyImageSearch. (n.d.). Retrieved June 8, 2024, from https://pyimagesearch.com/2021/05/14/convolutional-neural-networks-cnns-and-layer-types/

Cross Entropy Loss: Intro, Applications, Code. (n.d.). Retrieved June 7, 2024, from https://www.v7labs.com/blog/cross-entropy-loss-guide

CS231n Convolutional Neural Networks for Visual Recognition. (n.d.). Retrieved June 8, 2024, from https://cs231n.github.io/convolutional-networks/

Decoding Softmax: Understanding Its Functions and Impact in AI - Zilliz blog. (n.d.). Retrieved June 7, 2024, from https://zilliz.com/learn/decoding-softmax-understanding-functions-and-impact-in-ai

Difference Between a Batch and an Epoch in a Neural Network - MachineLearningMastery.com. (n.d.). Retrieved June 8, 2024, from https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/

Epoch vs Iteration when training neural networks? - Intellipaat Community. (n.d.). Retrieved June 8, 2024, from https://intellipaat.com/community/35/epoch-vs-iteration-when-training-neural-networks

How to Implement Softmax and Cross-Entropy in Python and PyTorch - GeeksforGeeks. (n.d.). Retrieved June 7, 2024, from https://www.geeksforgeeks.org/how-to-implement-softmax-and-cross-entropy-in-python-and-pytorch/

machine learning - Epoch vs Iteration when training neural networks - Stack Overflow. (n.d.). Retrieved June 8, 2024, from https://stackoverflow.com/questions/4752626/epoch-vs-iteration-when-training-neural-networks

Overfitting in Machine Learning: What It Is and How to Prevent It. (n.d.). Retrieved June 8, 2024, from https://elitedatascience.com/overfitting-in-machine-learning

Overfitting in Machine Learning and Computer Vision. (n.d.). Retrieved June 8, 2024, from https://blog.roboflow.com/overfitting-machine-learning-computer-vision/

Question about use of torch.max function to calculate accuracy - vision - PyTorch Forums. (n.d.). Retrieved June 8, 2024, from https://discuss.pytorch.org/t/question-about-use-of-torch-max-function-to-calculate-accuracy/187500

Softmax and Cross Entropy Loss. (n.d.). Retrieved June 7, 2024, from https://www.parasdahal.com/softmax-crossentropy

What is Overfitting? - Overfitting in Machine Learning Explained - AWS. (n.d.). Retrieved June 8, 2024, from https://aws.amazon.com/what-is/overfitting/

What is Overfitting? | DataCamp. (n.d.). Retrieved June 8, 2024, from https://www.datacamp.com/blog/what-is-overfitting

What is Overfitting? | IBM. (n.d.). Retrieved June 8, 2024, from https://www.ibm.com/topics/overfitting

What is the Difference Between "Epoch" and "Iteration" in Training Neural Networks - GeeksforGeeks. (n.d.). Retrieved June 8, 2024, from https://chjurnalites24.net/?_=%2Fwhat-is-the-difference-between-epoch-and-iteration-in-training-neural-networks%2Famp%2F%23KJWqMdlUlBn8PPpbVxjljJjzd5NnHlqquK0%2FPGA%3D