Computational Intelligence ( )

Submission Assignment 5

Instructor: JakubM. Tomczak *Name:* Begüm Yalçın, *Netid:* 2797623

1 Introduction

- The problem proposed in this assignment is to use an evolutionary algorithm to select the best architecture of a neural network. The assignment is a combination of assignment 3 and assignment 4. Similar to assignment 4, the neural network is trained on images. We aim to obtain a neural network architecture by using neural architecture search.
- The used methodology is basically creating a convolutional neural network architecture. The structure of the neural network is Conv2d → f(.) → Pooling → Flatten → Linear 1 → f(.) → Linear 2 → Softmax. For the components of the architecture, there are different options. We aimed to create phenotype representations for the different configurations. Also, genotype representations are created for these configurations. We will make use of the evolutionary algorithms as done in the assignment 3 , then combine the evolutionary algorithm structure with the neural network architecture. The neural architecture will be compared to the neural network architecture of assignment 4.

2 Problem statement

- In this assignment, we were assigned to use an evolutionary algorithm for selecting the best architecture of a neural network. The neural network strcuture should be in the form of Conv2d → f(.) → Pooling → Flatten → Linear 1 → f(.) → Linear 2 → Softmax. We have different choices in each building block: * Conv2d: Number of filters: 8, 16, 32 / kernel=(3,3), stride=1, padding=1 OR kernel=(5,5), stride=1, padding=2 * f(.): ReLU OR sigmoid OR tanh OR softplus OR ELU * Pooling: 2x2 OR Identity Average OR Maximum * Linear 1: Number of neurons: 10, 20, 30, 40, 50, 60, 70, 80, 90, 100.
- In total, there are 4500 possible configurations.
- The problem involves examining possible solutions and deciding on the best performing architecture . It is indeed to find the best combination of architectures and configurations. The fastest learning duration and the best prediction is the goal in this assignment. We used here evolutionary algorithms with a sample of population of size 7 and number of generations 11. The evolutionary algorithms tests the best combination of architecture and tries to find the best performing one. The best one is decided by examining the classification error.
- We have here as the objective function that minimizes the classification error and reduces complexity. So that could be objective = a * regularization + classification error. a is a constant for the training. The objective function is used to create efficient, less complex and better resulting algorithms. Possible difficulties could be that the training can take too long if the algorihtm is not efficient. This difficulty could be reduced by lowering the epoch number.

3 Methodology

- While doing the assignment, I got help from the previous assignments 3 and 4. I applied the Evolutionary Algorithm structure from assignment 3 and the Convolutional Neural Network architecture from assignment 4. Then I combined them to apply Neural Architecture Search.
- I first imported necessary libraries, then used the digits dataset from assignment 4. I also took flatten and reshape methods. I applied ClassifierNeuralNet class from assignment 4. I also took the evaluation and training functions from the previous assignment. Then I initialized dataloaders with batch size 64 from Digits dataset. I wrote genotype and phenotype representations as a dictionary. I created

dictionaries for Convolutional number of filters, convolutional layer configurations, pooling dimensions, pooling type, number of neurons in neural layer, and activation function. I created a CNNGenerator class with createpopulation, decode, createnetwork and chooseactivation functions.

- createnetwork creates a neural network from the individuals. choose activation chooses activation function according to key words. decode function decodes individuals representations from the values of the dictionaries. create population generates a population with random individiuals and uses lengths of dictionaries defined.
- Then I used the code from assignment 3 for the evolutionary algorithm. I also created an Evaluator class for the evolutionary algorithm. the Evaluator class does the evaluation for the Evolutionary Algorithm class. The evaluation is very similar to the evaluation part in the assignment 4. Then I initialized the EAEvaluator and ran it. Then I initislized the CNN model and trained it.

4 Experiments

- We created dictionaries for Convolutional number of filters, convolutional layer configurations, pooling dimensions, pooling type, number of neurons in neural layer, and activation function.
- We implemented a convolutional neural network of the following form: Conv2d $\rightarrow$ f(.) $\rightarrow$ Pooling $\rightarrow$ Flatten $\rightarrow$ Linear 1 $\rightarrow$ f(.) $\rightarrow$ Linear 2 $\rightarrow$ Softmax
- We have different choices in each building block:

  * Conv2d: Number of filters: 8, 16, 32 / kernel=(3,3), stride=1, padding=1 OR kernel=(5,5), stride=1, padding=2 * f(.): ReLU OR sigmoid OR tanh OR softplus OR ELU * Pooling: 2x2 OR Identity Average OR Maximum * Linear 1: Number of neurons: 10, 20, 30, 40, 50, 60, 70, 80, 90, 100
- The hyperparamaters of the experiment are: we have lambda penalty value of 0.01, number of generations 11, population size 7, learning rate 1e-3, weight decay 1e-5, number of epochs for the CNN model 300, maximum patience value 20. For optimizer we used ADAMAX and for the loss function we used negative log likelihood function.
- The experiments tried to prove what is the best way to have a neural network architecture. We tried to apply neural architecture search algorithm by combining evolutionary algorithms and Convolutional Neural Network architecture.

5 Results and discussion

- The individual used is[16, 'config0', 'ReLU', (2, 2), 'maximum', 100, 'ReLU']. So there are 16 convolutional filters, convolutional config config0 which is kernelsize=3, stride=1, padding=1, relu activation function, pooling dimensions (2,2), pooling type maximum, linear num neurons 100 and again ReLU activation function.
- The evolutionary algorithm is trained with 11 number of generations and population size of 7. The algorithm after trained and evaluated, gave a final result of 0.237 as classification error. The best fitness values started from 0.05 in the first generation then dropped to 0.04 and continued the same until the 11th generation.
- After the training and evaluation of the Convolutional Neural Network model, we have received a final classification error 0.081 which shows that the model gives nearly correctly results. The model was trained with 300 epochs. After 300th epoch, the model's classification error started to increase slightly. As to take best results, we had trained the model until 300th epoch. We applied a maximum patience rate of 20 and learning rate 1e-3. The classification error started from 0.314 in the first epoch then dropped to 0.081.
- The results show that CNN model gives better results compared to the evolutionary algorithm as it has lower classification error.
- Possible extension could be increasing the number of generations. In this way the fitness values would be bigger within generations. Also, the population size is too small which is 7, this could be increased in order to have better results.

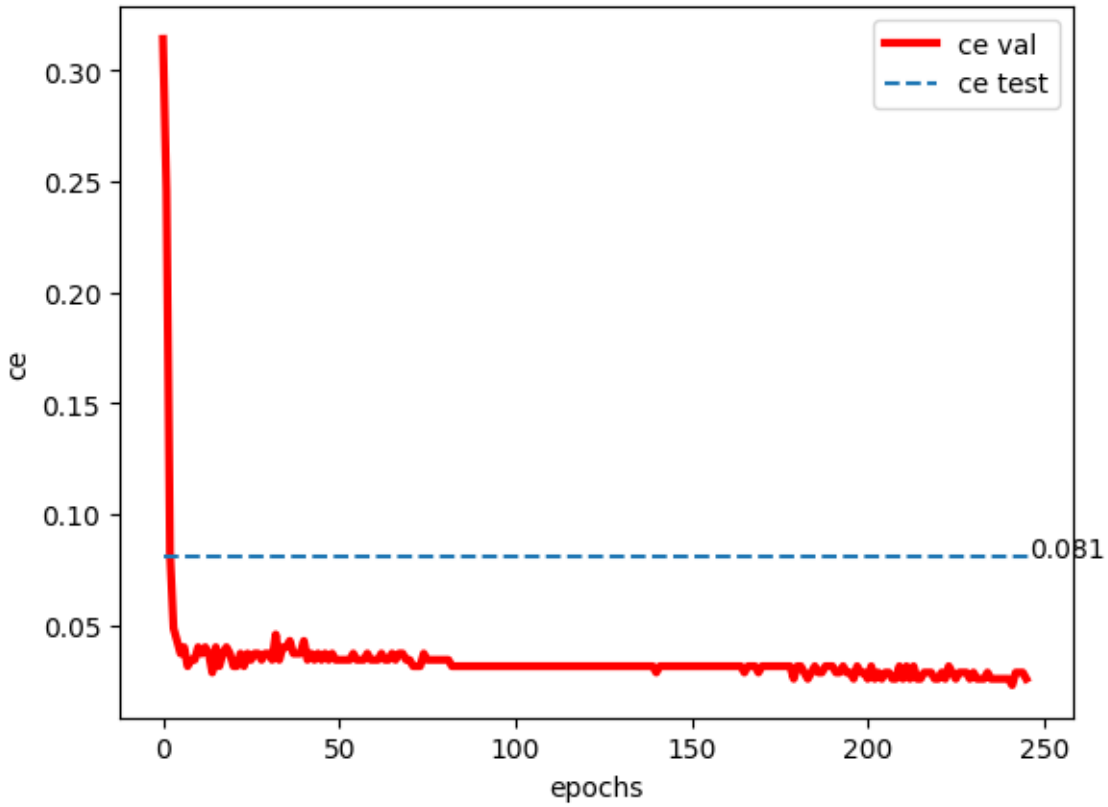

Figure 1: ce (classification error) val curve on test set

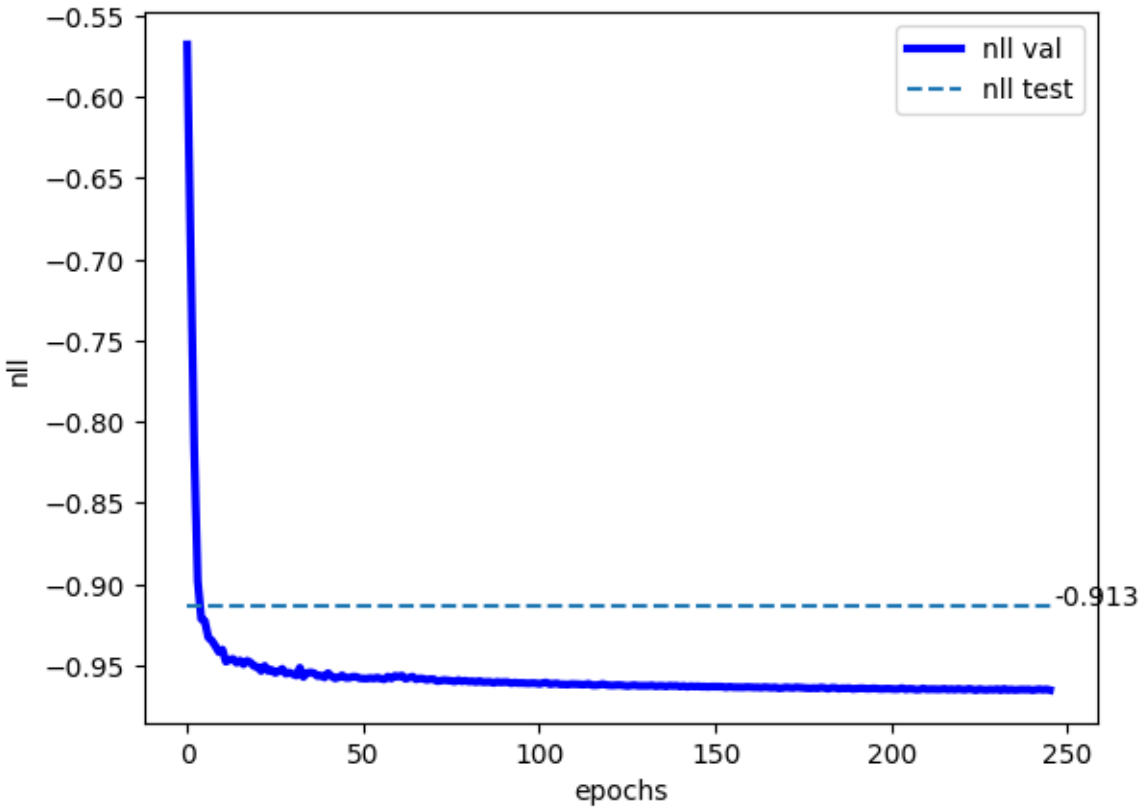

Figure 2: nll (negative log likelihood) val curve on test set

References

Indicate papers/books you used for the assignment. References are unlimited. I suggest to use `bibtex` and add sources to `literature.bib`. An example citation would be Eiben et al. (2003) for the running text or otherwise (Eiben et al., 2003).

References

Eiben, A. E., Smith, J. E., et al. (2003). *Introduction to evolutionary computing*, volume 53. Springer.