Neural Networks and Deep Learning 2024/25: Topics and Projects

To be studied

From "Neural Networks for Pattern Recognition"
Section 1: 1.1,1.2, 1.3, 1.4., 1.8,1.9
Section 3: 3.1, 3.2, 3.5
Section 4: 4.1, 4.2, 4.3, 4.8
Section 6: 6.1,6.6, 6.7, 6.9
Section 7: 7.1, 7.4, 7.5
Section 8: 8.1, 8.2,8.5

From "Deep Learning: Foundations and concepts"
Section 6: 6.2, 6.2.1, 6.2.2, 6.2.3, 6.3, 6.3.1, 6.3.2, 6.3.3
Section 10: 10.1, 10.2
Section 12: 12.1

Final Projects

A discussion supported by a presentation on one of the following projects

1. **(Low Difficulty)**

   Consider the raw images from the MNIST dataset as input. This is a classification problem with C classes, where C= 10. Extract a global dataset of N pairs, and divide it appropriately into training and test sets (consider at least 10,000 elements for the training set and 2,500 for the test set). Use standard gradient descent as the weight update algorithm. Study the learning process of a neural network (e.g., epochs required for learning, error trend on training and validation sets, accuracy on the test set) with a single layer of internal nodes, varying the learning modality: online, batch, and mini-batch. Conduct this study for at least three different dimensions (number of nodes) for the internal layer. Select and keep activation functions constant. If necessary, due to computational time and memory constraints, you can reduce the dimensions of the raw MNIST dataset images (e.g., using the `imresize` function in MATLAB).

2. **(Low Difficulty)**

   Consider the raw images from the MNIST dataset as input. This is a classification problem with C classes, where C= 10. Extract a global dataset of N pairs, and divide it appropriately into training and test sets (consider at least 10,000 elements for the training set and 2,500 for the test set). Use resilient backpropagation (RProp) as the weight update algorithm (batch update). Study the learning process of a neural network (e.g., epochs required for learning, error trend on training and validation sets, accuracy on the test set) with a single layer of internal nodes, varying the number of internal nodes (selecting at least five different dimensions) and using cross-entropy loss with soft-max. Select and keep all other hyperparameters constant, such as activation functions and RProp parameters. If necessary, due to computational time and memory constraints, you can reduce the dimensions of the raw MNIST dataset images (e.g., using the `imresize` function in MATLAB).

3. **(Low Difficulty)**

   Consider the raw images from the MNIST dataset as input. This is a classification problem with C classes, where C= 10. Extract a global dataset of N pairs, and divide it appropriately into training and test sets (consider at least 10,000 elements for the training set and 2,500 for the test set). Use gradient descent with momentum as the weight update algorithm. Study the learning process of a neural network (e.g., epochs required for learning, error trend on training and validation sets, accuracy on the test set) with a single internal layer of neurons, varying η (learning rate) and the momentum for at least five different dimensions (number of nodes) of the internal layer. Select and keep all other parameters constant, such as output functions. If necessary, due to computational time and memory constraints, you can reduce the dimensions of the raw MNIST dataset images (e.g., using the `imresize` function in MATLAB).

4. **(Medium Difficulty)**

Consider the raw images from the MNIST dataset as input. This is a classification problem with C classes, where C=10. Extract a global dataset of N pairs (consider at least 10,000 elements). Use resilient backpropagation (RProp) as the weight update algorithm, with a neural network having a single layer of internal nodes. Choose the model hyperparameters, i.e., the RProp parameters (η+) and (η-) and the number of internal nodes, based on a k-fold cross-validation approach (e.g., k= 10). Select and keep all other parameters constant, such as activation functions. If necessary, due to computational time and memory constraints, you can reduce the dimensions of the raw MNIST dataset images (e.g., using the `imresize` function in MATLAB).

5. **(Medium Difficulty)**

   Consider the raw images from the MNIST dataset as input. This is a classification problem with C classes, where C= 10. Extract a global dataset of N pairs, and divide it appropriately into training and test sets (consider at least 10,000 elements for the training set and 2,500 for the test set). Use resilient backpropagation (RProp) as the weight update algorithm (batch update). Study the learning process of a neural network (e.g., epochs required for learning, error trend on training and validation sets, accuracy on the test set) by varying the number of internal layers from 1 to 5. Compare the case where the activation function of the nodes is the hyperbolic tangent with cases where the activation function is ReLU and leaky ReLU. Test different choices for the number of nodes in the internal layers. If necessary, due to computational time and memory constraints, you can reduce the dimensions of the raw MNIST dataset images (e.g., using the `imresize` function in MATLAB).

6. **(Medium-High Difficulty)**

   Consider the raw images from the MNIST dataset as input. This is a classification problem with C classes, where C= 10. Extract a global dataset of N pairs, and divide It appropriately into training and test sets (consider at least 10,000 elements for the training set and 2,500 for the test set). Following the article *"Empirical evaluation of the improved RProp learning algorithms, Christian Igel, Michael Husken, Neurocomputing, 2003"*, compare the classic resilient backpropagation (RProp) with at least two proposed variants of the algorithm as weight update methods (batch update). Fix the activation function and the number of internal nodes (at least three different dimensions), and compare the results obtained with the different learning algorithms. If necessary, due to computational time and memory constraints, you can reduce the dimensions of the raw MNIST dataset images (e.g., using the `imresize` function in MATLAB).

7. **(Medium-High Difficulty)**

   Consider the raw images from the MNIST dataset as input. This is a classification problem with C classes, where C= 10. Extract a global dataset of N pairs (at least 10,000). Use resilient backpropagation (RProp) as the weight update algorithm, with a neural network having a single layer of internal nodes. Select the model hyperparameters, i.e., the RProp parameters (η+ and η-) and the number of internal nodes, based on a k-fold cross-validation approach (e.g., k= 10). Compare the classical "grid" approach with the "random" approach (*J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, 2012*) for hyperparameter search. Select and keep all other parameters constant, such as activation functions. If necessary, due to computational time and memory constraints, you can reduce

the dimensions of the raw MNIST dataset images (e.g., using the `imresize` function in MATLAB).

8. **(Medium-High Difficulty)**

   Consider the raw images from the MNIST dataset as input. This is a classification problem with C classes, where C=10. Extract a global dataset of N pairs, and divide it appropriately into training and test sets (consider at least 10,000 elements for the training set and 2,500 for the test set). Use resilient backpropagation (RProp) as the weight update algorithm (batch update). Study the learning process of a neural network (e.g., epochs required for learning, error trend on training and validation sets, accuracy on the test set) with a single layer of internal nodes and using the sigmoid activation function, varying the early-stopping criterion. Referring to the article *"Early Stopping — But When? Lutz Prechelt, 1999"*, consider the two early-stopping algorithms GL and PQ with different values for the parameter α\alpha. Study networks with a different number of internal nodes (at least five different dimensions). If necessary, due to computational time and memory constraints, you can reduce the dimensions of the raw MNIST dataset images (e.g., using the `imresize` function in MATLAB).

9. **(Medium-High Difficulty)**

   Consider the raw images from the MNIST dataset as input. This is a classification problem with C classes, where C=10. Extract a global dataset of N pairs, and divide it appropriately into training and test sets (consider at least 10,000 elements for the training set and 2,500 for the test set). Use resilient backpropagation (RProp) as the weight update algorithm (batch update). Construct k autoencoders, each one with a single internal layer composed of m_h nodes, where h=1,2,…,k and k=5 (for example m_1=25, m_2=50, m_3=75, m_4=100, m_5=125). This produces k encoders E_h, with h=1,2,…,k, allowing the data x to be projected into k different representations. For each encoder E_h, study the learning process of a neural network (e.g., epochs required for learning, accuracy on the test set) with a single layer of internal nodes and using the sigmoid activation function, taking as input the output of the encoder (E_h (x)

10. **(High Difficulty)**

    Consider the raw images from the MNIST dataset as input. This is a classification problem with C classes, where C=10. Extract a global dataset of N pairs, and divide it appropriately into training and test sets (consider at least 10,000 elements for the training set and 2,500 for the test set). Use resilient backpropagation (RProp) or ADAM as the weight update algorithm. Compare Convolutional and full-connected neural network performance at varying of the respective hyperparamenters, and considering the same weight update algorithm

NOTE THAT:
1) For all the projects you can either use a standard neural network library such as pytorch or develop a your own neural network library from scratch.
2) Each project can be developed by a group composed of up to 3 people