ISTANBUL TECHNICAL UNIVERSITY

COMPUTER ENGINEERING

Assignment 3

Artificial Intelligence (BLG 521E)

Jan 14th, 2022

Ahmet Kemal Yetkin

504201506

Problem 1 - Deep Learning

1. **Give two example loss functions that are used for neural network training, briefly explain their differences and for which problems they are preferred.**

   The loss function quantifies the difference between the expected outcome and the outcome produced by neural network. “Mean Squared Error” and “Categorical Cross-Entropy” are two examples of loss functions that are used for neural network training.

   - <u>Mean Squared Error (MSE)</u> is calculated as the average of the squared differences between the predicted and actual values. The result is always positive in MSE. The difference is a square that gives more importance to outliers. Therefore, MSE function is sensitive to outliers. If we had to predict one value for all targets, the prediction should be the mean. It can be used in the case of regression problems where a quantity is predicted and in regression settings where expected and predicted outcomes are real number values.
   - <u>Cross-Entropy</u> is commonly used in classification problems. It computes the difference between two probability distribution functions. The cross-entropy measures same thing that log loss measures. There some types of cross-entropy. It can be used with binary and multiclass classification problems. *Binary Cross-entropy* for binary classification problems and *Categorical Cross-entropy* for multiclass classification scenarios.

2. **Today, neural networks are used widely in machine learning problems because they can learn any function that performs a mapping between input data and the output data. Explain why non-linear activation functions must be used between neural network layers.**

   Activation functions are mathematical equations that determine the output of a neural network. Activation functions is used to introduce non-linearity into the network. Without a non-linear activation functions, only linear combinations are calculating in neural network.

   We are giving some input and generating some output in neural network. So, the given input may not be always linear. Non-linear activation functions must be used in order to bring in the much needed non-linearity property that enables them to approximate any function and for the given input that is not linear which is a complex input. They allow the model to create complex mappings between the network’s inputs and outputs.

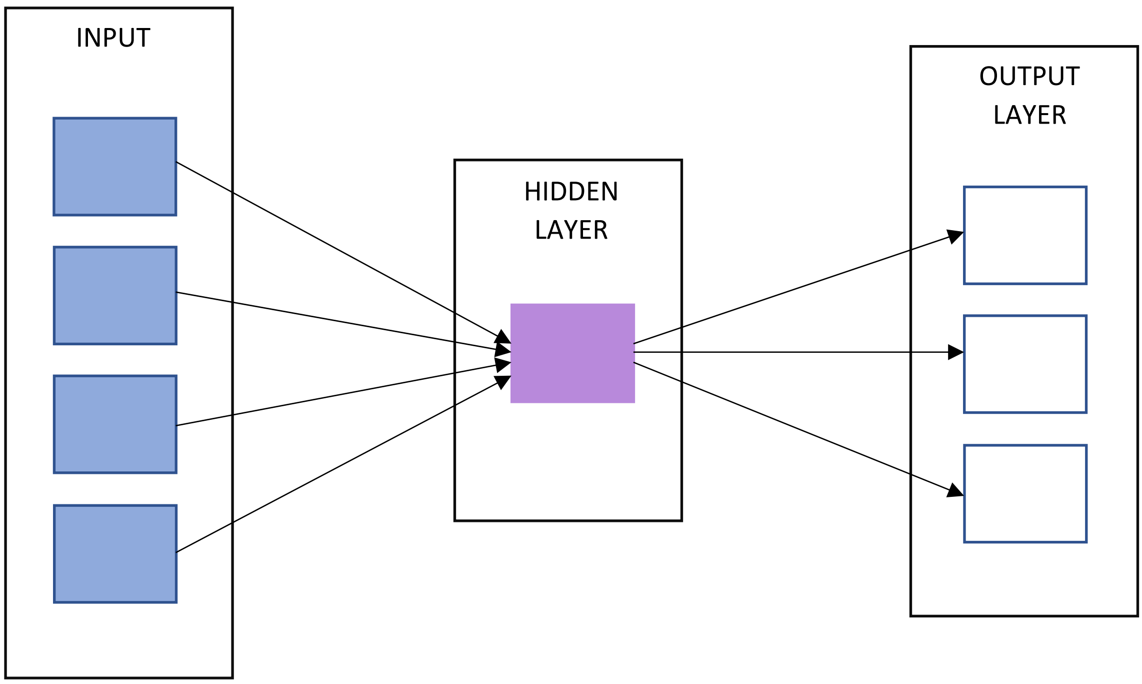
3. **Currently there are different neural network architectures being used. For both ”convolutional neural networks” and ”recurrent neural networks”, give an example**

domain/problem where these architectures are used and explain briefly why these specific architectures are preferred in this specific domain/problem.

Convolutional Neural Network (CNN) has one or more convolutional layers. They are used mainly for image processing, segmentation, and classification problems. Let's say there is a dataset and each image in this dataset is 28x28x1. For this input we will need 28x28=784 number of neurons. This can be work normal. However, if the size of image is 1000x1000, this means that we will need 10^6 neurons in input layer. This is huge number and computation will be ineffective. Therefore, the CNN is needed for this kind of cases. It is exracting the feature of image and converting it into lower dimension. It keeps the characteristics while converting.

Recurrent Neural Network (RNN) is the first algorithm that remembers its input. The Apple's Siri is an example for this architecture. The user give information while talking with Siri. When the user asks question about something, he/she will wait answer from the Siri. Since the RNN has internal memory, it can remember things about the input it received. Thus, it will be able to predict what's coming next. This is why RNNs preffered for in Apple Siri. In addition, Google Translate also the example for this arhictecture.

4. **Draw a simple fully-connected neural network that takes 4 input features as input, has 1 hidden layer between input and output layers and is able to classify between 3 different classes. Give an example loss function for this task and explain how output of your neural network is used to determine the class of your input (How is the numerical value you obtained in the last layer used for classification?).**



We can use Multi-class SVM loss function for this task. It is basically used to quantify how “good” or “bad” a given predictor is at classifying the input data points in a dataset. The way that it is determining the output class from the input is, we need to perform a sum over all of the categories except the true category. We need to sum over all the incorrect categories, then we need to compare the score of the correct category, and the socre of incorrect category. If the score of the correct category is greater than the score of the incorrect category, we will get a loss of zero. We need to sum this up over all of the incorrect categories for the image and this can give us the final loss. We need to do for all dataset. This is just an example for one image. The loss function is:

$$L_i = \sum_{j \neq y_i} max(0, s_j - s_{y_i} + 1)$$

	MONKEY	DOG	GIRAFFE	CAT	Losses	
	4.1	-3.2	1.2	-2.2	0	
	2.1	4.8	-1.7	-1.1	0	
	5.2	1.3	4.2	2.6	3	max(0, 5.2 – 4.2 + 1) = 3
	-2.4	-3.5	3.9	-1.2	6.1	max(0, 3.9 – (-1.2) + 1 = 6.1

Problem 2 - Reinforcement Learning Exploration

- **Epsilon-Greedy Exploration**
  Greedy exploration always takes the best action known to the agent. The main idea is that the agent should execute a random action with probability ε, whereas for the other 1 − ε of the cases greedily. The idea is:
  - Choose a greedy action
  - Choose a random action with probability ϵ.

  The advantages of this exploration strategys is very simple and light computation. The disadvantages are not quite systematic and inefficient. This strategy does not focus on unexplored regions. The epsilon-greedy, where epsilon refers to the probability of choosing to explore, exploits most of the time with a small chance of exploring. It can be used in maze problems. If we want to give an example for it: let’s say there some slot machines in front of the agent. If the agent wants to win money, the agent needs to play several times on those slot machines to find which one is giving more money. In short, epsilon-greedy strategy picks the current best option most of the time, but also sometimes it can pick a random option with a small number of probabilities which is epsilon. Also, it can be used in Monte Carlo Simulations.

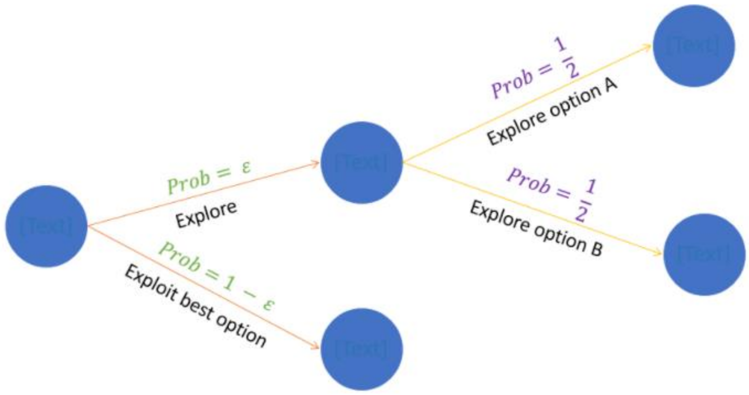


Figure 1. Epsilon-greedy

- **Thompson Sampling Exploration**

  The main ideas of Thompson (Posterior) Sampling are:
  - Sample MDP parameters from posterior distrubition
  - Act according to sample
  - Construct an optimal policy of the sampled MDP parameters
  - Update the posterior distrubition

The agent keeps track of a belief over the probability of optimal actions and samples from this distribution. Advantages of this strategy is systematic. It means it focuses more on unexplored regions. However, it is somewhat complicated and this is the disadvantage of the strategy. The example of this strategy can be wide range of structured decision problems and also shortest path problem.

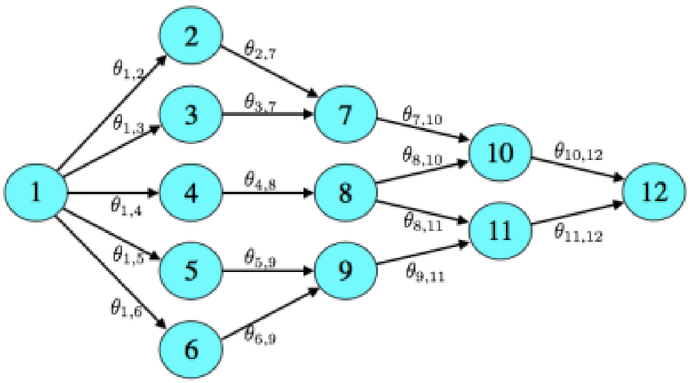


Figure 1.1: Shortest path problem.

- **Model-Based Exploration**
  In these problems, we are expecting from the agent that attempts to construct a model of how to effectively explore its environment. Essentially, the agent constructs an estimate of the way beneficial progressed understanding of a few a part of the environment will be to its performance. Methods in this section rely on external memory to resolve disadvantages of reward bonus-based exploration.
  Model-Based Reinforcement Learning methods learn a model of the domain for each state and action. The agent can then calculate a policy using this model through a method such as value iteration, effectively updating the Bellman equations for each state using its model. This is the example of the Model-Based Exploration strategy.

Problem 3 - Value Iteration on Grid

1. **What is the proper termination criteria for the value iteration algorithm?**

   Normally, value iteration requires an infinite number of iterations to converge. However, in the problem code it is given as 200 iteration as termination criteria.
   If we run the code for infinite number loop, it will not stop. Therefore, if want to add a termination criteria, we can stop once the value function changes by only a small amount in a sweep. By doing this, we will be able to compare with the previous state values. If the changes by only a small amount, it will terminate.

2. **If used algorithm would be policy iteration, what is the proper termination criteria for the policy iteration algorithm?**

   Policy iteration has clear termination criteria compared to the value iteration. In policy iteration, policy actions changes in every step. Thus, if there is no changes in policy actions, the algorithm terminates. It means once the policy is stable, it is will be optimal. This is the proper termination criteria of policy iteration algorithm.