

i Frontpage IN4050

IN4050 - Introduction to Artificial Intelligence and Machine Learning

Spring 2022

Tuesday, May 31

09:00 AM - 01:00 PM (4 hours)

All questions should be answered!

Each question is assigned a weight which is indicated.

The maximum number of points for the whole set is 100 points.

Permitted materials: None

An on-screen calculator is available.

You may answer in English, Norwegian, Danish or Swedish.

1 Search and Optimization

The search methods *gradient ascent/descent* and *hill climbing* are somewhat similar: Both work by taking a solution and iteratively change it a bit, moving step by step towards a local optimum. But there are also some important differences.

- Describe how gradient ascent and hill climbing differ with regards to how they change the current solution into the next solution (which is hopefully a bit closer to the optimum).
- For certain problems we can use hill climbing but not gradient ascent/descent. What kinds of problems are these? Why can't we use gradient ascent/descent there?

Suggested Solution

Max 3 points for a, 3 for b.

a) Gradient descent finds the next solution in the search by following the gradient, that is, the direction that the solution quality changes the most as we change the solution.

Hill climbing on the other hand typically finds the next solution by making a small, random change to the current solution - or testing several different random changes.

b) Gradient descent cannot be used for discrete (non-continuous) search/optimization problems. That is because they do not have a gradient: We cannot calculate the direction of largest change in a given point in the search landscape.

2(a) Evolutionary Algorithms - Representations

Briefly describe a problem where candidate solutions are best represented as permutations. Why are other representations less relevant/efficient for this problem?

2(b) Evolutionary Algorithms - Representations

Consider the solution [1 3 2 5 4] to a problem where solutions are permutations of the numbers from 1 to 5. Suggest a mutation operator that is appropriate for permutations and show an example of how it may transform the solution.

Suggested Solution

Max 3 points for a, 3 for b.

- a) Any solution most naturally represented as a permutation is ok here. That is, any problem where one needs to find an ordering of a given number of objects, such as the cities in a travelling salesman problem or tasks in a job scheduling problem. (1 p)
The second part asks why other representations are less efficient. The reason is that other representations are associated with different ways of changing solutions, thereby giving a less efficient search. For instance, if we represent a solution to a travelling salesperson problem as a list of integers (e.g. [1,2,3,4]) and treat it as a list of integers as we mutate solutions, we may mutate this list into [4,2,3,4] which is not a valid permutation. If we on the other hand treat it as a permutation and mutate it with the associated operators, we guarantee that we change it into a valid solution, and have a more efficient search. (2p)
- b) Many possible operators are given in the lecture on Evolutionary Algorithms. A simple one could be swap mutation: Swap 2 random elements from the solution. [1 3 2 5 4] could become [1 4 2 5 3].

3(a) Exploration and Exploitation in EA

What do we mean by the terms “exploration” and “exploitation” in EAs (and other search methods)?

3(b) Exploration and Exploitation in EAs

Name a way to change an EA so that it explores more

3(c) Exploration and Exploitation in EAs

What do we mean by premature convergence in an EA? How is premature convergence related to exploration/exploitation?

Suggested Solution

Max 3 points for each task.

For the given tasks, I am writing quite long answers for completeness. Students would not have to write all this for a full score – a brief explanation is enough.

- a) By “exploration” we mean trying new solutions very different from what we have tried before. This is important to avoid getting stuck in a local optimum and for properly evaluating a large part of the solution landscape.

By “exploitation” we mean taking solutions we know to be good and changing them a little bit, exploiting the knowledge we already have about what is working well. This is important to ensure the search can gradually improve solutions, avoiding that we are skipping randomly around in the solution landscape.

Search algorithms typically try to find a balance between these two for an efficient search that does not get stuck in local optima but that is still able to gradually improve promising solutions.

- b) There are many solutions here. Mostly they would be concerned with increasing the randomness in the EA. Examples: Increasing the mutation rate, increasing the impact of mutations (that is, how “strong” each mutation is), increasing the probability of crossover, decreasing the selection pressure by having more randomness in selection. Fitness sharing or other diversity maintenance techniques (crowding, speciation or island models) could also be used here to ensure solutions that are different from others are given preference.

- c) Premature convergence means that the solutions in the population too early stop exploring and get centered on the most promising peak(s) in the fitness landscape. This is caused by having a too strong selection pressure, leading to too much exploitation and too little exploration.

4 **k Nearest Neighbors (kNN)**

Describe shortly the main steps in the k nearest neighbors algorithm for supervised classification. (Answer 1-2 sentences for each step.)

Fill in your answer here

Maximum marks: 7

Suggested solution

1. Calculate the distance to all the training instances
2. Pick the k nearest ones
3. Choose the majority class for this set

Subtract 2 points for each missing point, and 1 point for a partially correct point.
1 point for the overall.

5(a) Linear regression predict

We consider a model for linear regression. We have trained a model with the weights $\mathbf{w} = (w_0, w_1, w_2) = (0, -1, 1)$. Each observation $\mathbf{x}$ has two features x_1 and x_2 . In addition, we have added one bias feature in the first position, $x_0 = -1$. Which value will this model predict for the observation $\mathbf{p} = (p_0, p_1, p_2) = (-1, 3, 1)$?

Fill in your answer here

Maximum marks: 3

Suggested solution

$$y = \mathbf{w} \cdot \mathbf{p} = (0, -1, 1) \cdot (-1, 3, 1) = -2$$

One point for the formula, one for entering the numbers correctly, one for the calculation.

5(b) Linear regression training

We assume that the model is trained sequentially by gradient descent and that it is updated after each observation. The target value for $\mathbf{p}$ is 0. What are the new weights after being trained on $\mathbf{p}$? Assume a learning rate of 0.1. Make additional assumptions if needed, and state them clearly. In particular, which loss function do you use?

Fill in your answer here

Maximum marks: 5

Suggested solution

$$\begin{aligned}\mathbf{w} &= \mathbf{w} - \eta(y - t)\mathbf{p} = (0, -1, 1) - 0.1(-2 - 0)(-1, 3, 1) = \\ &= (0, -1, 1) + 0.2(-1, 3, 1) = (-0.2, -0.4, 1.2)\end{aligned}$$

The loss function is (strictly speaking) 0.5 square error: $0.5(y - t)^2$

Up to 3 points for the new weights: one for the formula, one for entering correct numbers, one for calculation. Up to 2 points for loss function – only one point if naming the function without formula.

6(a) Multinomial logistic regression

We will use multinomial logistic regression to classify types of fruit. There are three different classes, and we have represented them as follows:

- nectarine: [1,0,0]
- apricot: [0,1,0]
- peach: [0,0,1]

To make it simple, there are only two features for each observation, x_0 and x_1 , where $x_0 = -1$ is a bias feature. The following is one particular observation,

$$\mathbf{x} = (x_0, x_1) = (-1, 0.5)$$

We have so far the following weight matrix,

$$\begin{bmatrix} w_{0,1} & w_{0,2} & w_{0,3} \\ w_{1,1} & w_{1,2} & w_{1,3} \end{bmatrix} = \begin{bmatrix} 0.7 & -0.5 & 0 \\ -0.2 & 0.4 & 0.6 \end{bmatrix}$$

The weight $w_{i,j}$ connects feature i to class j . How will the model classify the observation $\mathbf{x}$? State reasons for your answer.

Fill in your answer here

Maximum marks: 7

Solution

Write $W[:,j]$ for column j in the weight matrix. Then we get the weighted sums

$$z_1 = \mathbf{x} \cdot W[:,1] = (-1, 0.5) \cdot (0.7, -0.2) = -0.8$$

$$z_2 = \mathbf{x} \cdot W[:,2] = (-1, 0.5) \cdot (-0.5, 0.4) = 0.7$$

$$z_3 = \mathbf{x} \cdot W[:,3] = (-1, 0.5) \cdot (0, 0.6) = 0.3$$

We take the exponentials

$$u_1 = e^{z_1} = e^{-0.8} = 0.449$$

$$u_2 = e^{z_2} = e^{0.7} = 2.014$$

$$u_3 = e^{z_3} = e^{0.3} = 1.350$$

The sum of the exponentials $s = \sum_{i=1}^3 u_i = 3.813$

The predicted values

$$y_1 = \frac{u_1}{s} = 0.118$$

$$y_2 = \frac{u_2}{s} = 0.528$$

$$y_3 = \frac{u_3}{s} = 0.354$$

Hence we see that y_2 is largest and the item is classified as *apricot*.

Comments:

1. 7 points for correct
2. 6 points for everything correct except the calculations
3. 3 points if only taken the sums and conclude correct class from that
4. up to 5 points for partially correct recipe for y , e.g. taking exponents without normalizing or normalizing without exponents
5. a solution which only calculates the weighted sums and argues correctly that this suffices since both the exponential function and the softmax are monotone, can get a full score

6(b) Training multinomial logistic regression

We are training the classifier sequentially. The observation $\mathbf{x}$ is part of the training set with the target class *apricot*. What will the new values of the two weights $\mathbf{w}_{0,3}$ and $\mathbf{w}_{1,2}$ be after training on this observation? Use 0.2 as learning rate.

Fill in your answer here

Maximum marks: 5

$$w_{0,3} = w_{0,3} - \eta(y_3 - t_3)x_0 = 0 - 0.2 * (0.354 - 0) * (-1) = 0.0708$$

$$w_{1,2} = w_{1,2} - \eta(y_2 - t_2)x_1 = 0.4 - 0.2 * (0.528 - 1) * 0.5 = 0.4472$$

Comment: subtract one point for errors in calculations, up to two points for errors in formulas (indices) and up to two points for error in the y -s and t -s. This must be seen in relationship to how the errors in (6a) have been evaluated.

7(a) Activation in hidden layer

Which sort of problems could we have solved with a multi-layer perceptron (MLP) if we did not include an activation function in the hidden layer?

Fill in your answer here

Maximum marks: 3

Suggested solution

Linear regression and classification with linear decision boundaries. Adding hidden layers without an activation does not increase the computational power compared to no hidden layers.

7(b) **Batch vs SGD**

When training a classifier by gradient descent, we distinguish between batch training and stochastic gradient descent. Explain the differences between the two. Explain also what we mean by mini-batch training.

Fill in your answer here

Maximum marks: 6

Suggested solution

Batch training: Given a set of weights, one predicts the output values for all items in the training set. One then computes the loss and the gradient for this whole set and upgrade the weights accordingly. One then repeats this process a number of times.

Stochastic gradient descent: One calculates the loss and gradient for one item in the training set at a time, and repeat. *Stochastic* indicates that the next item to process is chosen randomly. This is in contrast to a sequential procedure where one goes through the data set a couple of times in a set order.

Mini-batch is an in-between the two others. One first decides on a batch size, for example 8 datapoints. The whole dataset is split into such mini-batches. In one epoch, one processes all the mini-batches. For each mini-batch, one calculates output, loss and gradient and update the weights accordingly. One then proceeds to the next mini-batch. One might consider shuffling the data points between the epochs.

Comments: Roughly 2 points for each of the three. We didn't talk much about the stochastic (random) part in the lectures, but it should be mentioned; one point for describing a sequential training procedure without mentioning the randomness. Also, it was detected a mistake in the use of the term *epoch* in one slide from the lectures, so be liberal with respect to how the term *epoch* is applied. It should also be taken into consideration that some texts uses SGD also for mini-batches (with randomness).

8 Random forest

According to Marsland, the main idea behind the random forest classifiers "is largely that if one tree is good, then many trees (a forest) should be better, provided that there is enough variety between them."

Which methods can be applied to provide variety between the trees?

Fill in your answer here

Maximum marks: 7

Suggested solution

One way to make variation between the trees is to train them on different bootstrap sets. Suppose the training set X contains n many datapoints. A bootstrap set, X_i , is constructed by picking n items from X with replacement. Then the set X_i will resemble the original set but normally not be identical to X . We may construct as many bootstrap sets as we like and train a decision tree classifier on each set.

Another way to make variation between the trees is to restrict the construction at each node to only consider a subset of the all features.

After training the random forest, we will have a collection of many decision trees that make independent decision. We get a classifier by taking the majority vote between the trees.

Comment: Up two three points for each of the two methods and one point for putting it together. For example, a solution which has only one of the two methods for making variation, but a correct description of how several trees can be used for voting: 4 points.

9 Recurrent neural networks

In Marsland's book (p. 71), we can read that recurrent neural networks (RNNs) "have been studied, but are not commonly used." This has changed a lot since the book was published. What is the main difference between recurrent neural networks and feed-forward networks (MLPs)? For which kind of problems are RNNs particularly suited?

Fill in your answer here

Maximum marks: 7

Suggested solution

A recurrent neural network (RNN) is a network which contains a cycle within its network connections. It may be a connection from a node to itself. It could also be a cycle including more nodes.

The RNN is processed in time steps. The value of a node x at time $t+1$ is calculated from the values of all the nodes at time t and the direct connections into node x . In particular, if there is a connection from a node x into itself, the state of x at time $t+1$ will be influenced by the state of x at time t (and all earlier states). This gives the network a form of memory.

An RNN is particularly suited for sequence problems, for example sentences in natural language, where it reads one word at a time and "remembers" the earlier words in the sentence.

Comment: Up to 3 points for cycles, 2 points for the connection to processing, and 2 points for sequence problems.

10	State/action	Q-value
	State S1, action A1	0.4
	State S1, action A2	0.4
	State S2, action A1	0.0
	State S2, action A2	0.8

(a) Learning Value Functions

For the next questions, consider a reinforcement learning problem with two states and two possible actions. We are trying to learn a table of Q-values representing the utility of performing each action in each state. At some point in our learning, the table looks like shown on the left.

An alternative to learning Q-functions is to directly learn estimated values (V) for each state. Referring to the Q-table on the left, name one disadvantage of doing this.

(b) V-values - Advantages

In the last task, you suggested a disadvantage to estimating state-values (V). What may be the *advantage* of estimating state-values (V) instead of values for state/action pairs (Q)?

Note, this is a general question on Reinforcement Learning. Here, you do not have to refer to the given Q-table.

(c) Q-learning

Assume we are in state S1 and perform action A2. We observe a reward of 1 and notice that we next find ourselves in state S2.

Calculate the updated Q-value $Q(S1, A2)$ assuming a discount factor of 0.9 and learning rate of 0.1. The update function for q-learning is given below, taken from the course textbook (Marsland). Show all steps of your calculation.

$$Q(s, a) \leftarrow Q(s, a) + \mu(r + \gamma \max_{a'} Q(s', a') - Q(s, a))$$

(d) SARSA

How would the calculation above change if you updated $Q(S1, A2)$ with SARSA instead of Q-learning?

What information are you missing to perform the SARSA-update?

Suggested Solution

Max. 3 points for each question

- a) Q-values give an estimate of the utility of performing a given action in a given state. V-values on the other hand estimate the utility of the state itself. A disadvantage of doing so is that this averages out the effects of different actions in that state, leading to a less informative estimate. In the table above, for instance, we see that State S1 and State S2 are very different: One has a medium reward for both actions, the other can give us a high reward if we choose A2, and a low reward for A1. This information would be hidden if we only knew the state-values instead of the Q-values.
- b) The advantage of learning state-values (V) instead of state-action values (Q) is that it requires less data. Imagine we have a problem with very many possible actions (e.g. 100) and 100 different states. That gives 10.000 (100*100) different state, action values we need to estimate, which would require very much data. Learning the state-values (V) directly however, would only require to estimate the 100 state-values.

- c) We use the Q-function as stated:

$$Q(s, a) \leftarrow Q(s, a) + \mu(r + \gamma \max_{a'} Q(s', a') - Q(s, a))$$

The old Q-value $Q(s,a)$ is 0.4. r is 1. The learning rate (μ) is 0.1. The discount factor (γ) is 0.9. The final part of the equation is the maximum future q-value, which is the maximum q-value in the new state we arrive in (S2). This is 0.8

The new q-value is thus $Q(S1, A2) = 0.4 + 0.1 \cdot (1 + 0.9 \cdot 0.8 - 0.4) = 0.532$

- d) The formula for Q-learning assumes we will take the optimal action in the new state we arrive in (that is, that we will choose action A2 when we get to state S2). Therefore, the estimated future Q-value that q-learning uses in its update is 0.8.

SARSA updates the Q-function in the same way, but rather than assuming that the optimal action will be chosen, it rather looks at what action was actually chosen. Therefore, if the agent chose to do the optimal action A2 in the new state S2, the calculation would not change. If the agent instead chose to perform A1 in the new state, the future q-value in the calculation above would change from 0.8 to 0.0.

We are therefore missing information about the action the agent chooses next (after getting to state S2) to perform the SARSA update.

11 Developmental Systems

How can developmental systems generate complex patterns from simple rules? Give a simple example, referring to one of the developmental systems discussed in this course's lecture.

Suggested Solution

Max. 5 points

Developmental systems contain “recipes” for transforming some input to an output. If we apply such a recipe to the input, and next use the output as the new input, we can gradually grow a small seed into a more and more complex structure.

An example is L-systems, which describe how to transform strings by applying a set of rules that transform the letters in these strings to new letters. For instance, if we have the initial string “A” and the rules “A→AB” “B→A”, then the initial string would after two developmental steps grow like this:

A

AB

ABA

The lecture on developmental systems contains many other examples, which would also be valid replies here if described correctly.

12 Hierarchical Clustering

We have performed 20 different blood tests on each of n patients and have collected all the results in a $20 \times n$ matrix $\mathbf{X}$ such that $\mathbf{X}[i,j]$ is the result of the i 'th blood test performed on the j 'th patient.

We wish to see if the patients cluster into subgroups based on the blood test results, and decide to use hierarchical clustering. **What distance measure and linkage method would you choose in each of these cases?**

Case 1: We wish to cluster together patients with quite similar blood test results (i.e. the distance between two patients is small if their results for each blood test are numerically quite similar). Furthermore, every element of a cluster should have a small distance to every other element of the same cluster.

Case 2: We wish to cluster together patients with correlated blood test results (i.e. the distance between two patients is small if their results are low on the same blood tests and high on the same blood tests). Furthermore, each element of a cluster should have a small distance to at least one other element of the same cluster.

Suggested Solution

Max. 6 points

Case 1: Manhattan distance or Euclidean distance would both work (or any other distance function $d(x,y)$ such that $d(x,y)$ is small if and only if x and y are numerically similar), but not cosine distance or other correlation measures. Since we want all elements in a cluster to be close to all other elements of the same cluster, we should choose complete linkage.

Case 2: Cosine distance or any other correlation measure would work, but not Manhattan distance or Euclidean distance. Since we require each element of a cluster to be close to at least one other element of the same cluster, we should choose single linkage.

13 Future Perspectives: Causality

- What do we mean when we say machine learning models are good at learning correlations, but not causal relationships?
- Give an example of a situation where a learned correlation is not enough to make a good decision, but where we also need information about causal relationships.

Suggested Solution

Max 3 points per question

- With supervised learning, we can easily learn relationships between input data and output data, including correlations between them. For instance, we could detect from data from a store selling ice cream that ice cream sales are generally higher when the weather is nice – by detecting this correlation between weather type and sales. However, these learned correlations do not tell us anything about causation (which action/event causes another to happen). From the point of view of the data and the machine learning model, it might as well be that ice cream sales are somehow having some effect on the weather.
- Situations where we want to intervene often require us to have some information about causality. For instance, imagine we learn from hospital records that patients that go more frequently to the doctor more frequently have a serious illness than those who do not. From this learned correlation, we may think we should try to make patients go less to the doctor to stay healthier. Knowing about the causal relationship (that in fact it is not the doctor that makes you sick, but you being sick that makes you see the doctor) helps us decide this is not a good idea.