MACHINE LEARNING

Test - 8 January 2021

---

1 Checkbox questions

1. (3 p.) Which one of the following is a regression problem?

   ☒ Perform stock market prediction on the basis of a window of previous samples;

   ☐ Decide whether a given fingerprint belongs to a given person;

   ☐ Predict whether the rank of a given Web page exceeds a given threshold;

   ☐ Nothing of the above.

2. (3 p.) Which of the following statements concerning the *one-hot* encoding is correct?

   ☐ The one-hot encoding corresponds with the traditional binary encoding of integers;

   ☐ The one-hot encoding is used only for deep networks;

   ☐ The one-hot encoding is based on one output only;

   ☒ The one-hot encoding of n classes consists of n outputs. The target is null for all outputs apart from the one which encodes the specific class.

   ☐ Nothing of the above.

3. (3 p.) Which one of the following is correct concerning the notion of training set and test set?

   ☐ Both the sets can be used for the discovery of the weights of the neural network;

   ☐ The test set can be used in the learning algorithm only to check overfitting;

   ☐ Training and test set are synonyms.

   ☒ The test set cannot be used in the computation of the weights of the neural network.

   ☐ Nothing of the above.

4. (3 p.) Let us consider the quadratic loss function

   $$V(f(x), y) = \frac{1}{2}(f(x) - y)^2$$

   Which of the following statements is correct?

   ☒ The loss function is null only if the output of the function fits perfectly the target;

   ☐ The loss function can be used for classification but not for regression;

   ☐ When we use sigmoidal units, this loss function cannot be used if the target y does not take values in $\{-1, +1\}$;

   ☐ Nothing of the above.

5. (3 p.) Let us consider the loss function

$$V(f(x), y) = \min\{0, 1 - y(x)f(w, x)\}$$

Which of the following statements is reasonable?

- ☒ The loss function is adequate for classification;
- ☐ The loss function is adequate for regression;
- ☐ The is not a loss function since it is not differentiable in all points of its domain;
- ☐ Nothing of the above.

6. (3 p.) What is the difference between loss function and empirical risk function?

- ☐ They are synonyms;
- ☒ The loss function refers to the error on single examples, whereas the risk function refers to the error over all the examples of the training set;
- ☐ The loss function is always differentiable whereas the empirical risk function may not be differential;
- ☐ Nothing of the above.

2 Open questions

1. (4 p) Discuss the following statements concerning the recognition performance of neural networks for handwritten chars.

   (a) If we significantly increase the 28×28 MNIST resolution we expect to increase significantly the recognition performance;

   (b) Pictures of handwritten digits that can be collected with ordinary smartphones have a resolution which is significantly higher than 28×28. Can you still see any other reason for keeping a limited resolution in the experiments with neural nets?

   (c) Suppose you have trained successfully a neural network on the MNIST database and you want to write an application which recognizes digits by using your own smartphone. Describe a pre-processing algorithm that uses the neural network trained on MNIST for recognizing the digits given by pictures taken on your smartphone at higher resolution.

(a) No, it is not the case that we can expect a significant increment of the performance, since we already achieve performances that are close to 100%.

(b) Of course! There is no need to use higher resolution for such a learning task, since the given internal representation of the chars is already very good for achieving top level discrimination of the classes (see also answer to (a)).

(c) Suppose you acquired your chars with resolution $m \cdot n$. We need to map $x \in \mathbb{R}^{m,n}$ onto $z \in \mathbb{R}^{28,28}$. We perform a sub-sampling based on gridding the picture at high resolution, where the grid is the one used for MNIST. The generic "box" is composed of $\lfloor m/28 \rfloor$ rows and $\lfloor n/28 \rfloor$ columns with $\lfloor m/28 \rfloor \cdot \lfloor n/28 \rfloor$ pixels. The equivalent input to be applied is

$$z_{\alpha,\beta} = \frac{1}{\lfloor m/28 \rfloor \cdot \lfloor n/28 \rfloor} \sum_{i=1}^{\lfloor m/28 \rfloor} \sum_{j=1}^{\lfloor n/28 \rfloor} x_{\lfloor m/28 \rfloor \alpha + i, \lfloor n/28 \rfloor \beta + j},$$

where $\alpha, \beta \in [0, 27]$.

This can be computed by the following algorithm:

$p \leftarrow \lfloor m/28 \rfloor$; $q \leftarrow \lfloor n/28 \rfloor$;
$m \leftarrow p \cdot q$;
for $a \leftarrow 0$ ***to*** 27 **do**
  for $b \leftarrow 0$ ***to*** 27 **do**
    for $i \leftarrow 0$ ***to*** $p - 1$ **do**
      for $j \leftarrow 0$ ***to*** $q - 1$ **do**
        $z[a, b] = z_{\lceil}a, b] + x[p \cdot a + i, q \cdot b + j]$
      end
    end
  end
end

2. (4 p) Let us consider the problem of predicting the number of deaths due to covid-19 on the basis of the collection $\mathcal{D} = \{(\kappa, y_\kappa)\}$, where κ denotes the day, expressed as an integer number, from the beginning of the outbreak. Suppose we are at the beginning of the outbreak and that the following approximations f_κ are good candidates for approximating y_κ:

$$f_\kappa = a_2\kappa^2 + a_1\kappa + a_0$$
$$f_\kappa = \alpha\beta^\kappa.$$

Show how can we convert these regression problems in linear regression.

Let us begin with the polynomial function. We set

$$p := \kappa^2$$
$$q := \kappa$$

so as

$$f_\kappa = \tilde{f}_{p,q} = a_2 p + a_1 q + a_0. \tag{1}$$

In the case of the exponential function we have

$$\log f_\kappa = \log \alpha + \kappa \log \beta$$

Now we set

$$y_\kappa := \log f_\kappa$$
$$m := \log \beta$$
$$q := \log \alpha$$

Finally, like for Eq. 1 we reduce

$$y_\kappa = m \cdot \kappa + q \tag{2}$$

which allows us to perform to linear regression as required.

3. (4 p) Consider the regression problem for estimating the following variables:

   - acceleration $0 - 100 Km/h$;
   - maximum velocity;
   - consumption at 90 Km/h

   on the basis of the following inputs

   - power (Kw)
   - weight (Kg)
   - drag coefficient c_x

   The training set is composed of 1500 examples of cars. Suppose that we want to enforce a stopping criterion for learning based on the percentage error for each output that must be kept below 5%. Suppose we use the following empirical risk:

$$E = \sum_{\kappa=1}^{1500} (y_\kappa - f(w, x_\kappa))^2.$$

   Determine the threshold ϵ of the stopping criterion $E < \epsilon$ under the above condition on the percentage error.

Let $a_\kappa^m, v_\kappa^m, c_\kappa^m$ be the minimum values on pattern κ of the three outputs (acceleration, maximum velocity, consumption). The condition on percentage error can be expressed by

$$\delta_{\kappa,i} = 100\frac{|y_{\kappa,i} - f(w, x_\kappa)|}{|y_{\kappa,i}|} < 5,$$

where $i = 1, 2, 3$ is the index which identifies the three outputs. Hence

$$|y_{\kappa,i} - f(w, x_{\kappa,i})| = \frac{1}{20} \cdot |y_{\kappa,i}|.$$

As a consequence, we can translate this condition on the empirical risk as follows

$$E = \sum_{\kappa=1}^{1500}\sum_{i=1}^{3}(y_{\kappa,i} - f(w, x_{\kappa,i}))^2 = \frac{1}{400}\sum_{\kappa=1}^{1500}\sum_{i=1}^{3}\cdot|y_{\kappa,i}|^2.$$

The most restrictive condition is the one on the minimum values $y_{\kappa,i}^m$ of the three outputs $a_\kappa^m, v_\kappa^m, c_\kappa^m$, so as we can choose

$$\epsilon = \frac{1}{400}\sum_{\kappa=1}^{1500}\sum_{i=1}^{3}\cdot|y_{\kappa,i}^m|^2$$

Remark: An important remark is in order which concerns the definition of the percentage error and its adoption. Clearly, it makes sense when the target and the corresponding value of the function is far from null values. As $y_{\kappa,i} \to 0$ we are in front of an ill-posed definition!

4. (Optional. 6 p) Let us consider the neural networks used for learning the `XOR` predicate where where we have 2 hidden units and one output. Suppose that the units are numbered beginning from $\kappa = 0$ as follows: inputs $(0, 1)$, hidden units $(2, 3)$, output units 4 and suppose that

$$w_{4,2} = w_{4,3} = 2, \quad b_4 = b_3 = b_2 = 1$$
$$w_{2,0} = w_{3,0} = 3, \quad w_{2,1} = w_{3,1} = 1,$$

Discuss the evolution of learning from this configuration.

This is in fact a configuration which on which the Backpropagation learning algorithm gets stuck. This can promptly be seen from the symmetric structure of the configuration. Neurons 2 and 3 share the same weights in the connection to the input. As a consequence the evolution of learning is equivalent to the one of a neural network with only one hidden unit as follows

$$f(x_0, x_1) = \sigma(2w_{4,2}\sigma(w_{2,0}x_0 + w_{2,1}x_1 + b_2) + b_4).$$

This function only construct linearly-separable configurations in (x_0, x_1). Suppose $\sigma(\cdot) = \tanh(\cdot)$. This comes from exploring the condition

$$2w_{4,2}\sigma(w_{2,0}x_0 + w_{2,1}x_1 + b_2) + b_4 = 0.$$

If $w_{4,2} \neq 0$ then this corresponds with

$$w_{2,0}x_0 + w_{2,1}x_1 + b_2 + \sigma^{-1}\left(\frac{b_4}{2w_{4,2}}\right) = 0,$$

which indicates that the separation condition is the one of the common hidden neurons 2,3. Clearly the remaining case $w_{4,2}$ is trivial.