Iniziato	venerdì, 17 febbraio 2023, 09:16
Stato	Completato
Terminato	venerdì, 17 febbraio 2023, 10:17
Tempo impiegato	1 ora 1 min.
Valutazione	**16,00** su un massimo di 20,00 (**80**%)
Feedback	Thanks for your attempt, all the best!

Domanda **1**

Parzialmente corretta

Punteggio ottenuto 1,50 su 2,00

Pretend you are a seasoned neural network expert, which of the following is a proper statement you should be supposed to say?

True	False		
○ ✗	◉ ☑	I prefer RNN to LSTM, they have less parameters to train and do not suffer vanishing gradient as well	✔
◉ ✗	○ ☑	I prefer small sized batches, down to single sample ones, because this way I reduce the overall computation and speed up training, especially on GPUs	✖
○ ✗	◉ ☑	I always set the initial hidden state of a RRN / LSTM to zero when doing inference because there is no way to know what should be the initial state for any specific dataset.	✔
◉ ☑	○ ✗	Choosing among the different approaches presented in class, I prefer early stopping to prevent overfitting because provided I have enough data for the validation step and a rough estimate of patience is sufficient in most of the cases it has no hyperparameter to tune	✔

I prefer RNN to LSTM, they have less parameters to train and do not suffer vanishing gradient as well: False

I prefer small sized batches, down to single sample ones, because this way I reduce the overall computation and speed up training, especially on GPUs: False

I always set the initial hidden state of a RRN / LSTM to zero when doing inference because there is no way to know what should be the initial state for any specific dataset.: False

Choosing among the different approaches presented in class, I prefer early stopping to prevent overfitting because provided I have enough data for the validation step and a rough estimate of patience is sufficient in most of the cases it has no hyperparameter to tune: True

Domanda **2**

Risposta corretta

Punteggio ottenuto 2,00 su 2,00

You are training an FFNN on some classification task, but it seems not to learn at all. After a few initial epochs (e.g., 2 or 3) where the loss function decreases, but still remains extremely high, you observe a (very) long series of epochs without any significant decrease or increase in the loss function. I.e., you get a flat line with little brittles here and there for both training and validation which are basically overlapped.

Which of the following are reasonable actions a neural network expert will do in a situation like this to fix or investigate the causes of the possible issue?

Notes:

1. All statements are independent of each other and thus they should be considered in isolation when selecting them
2. A neural network expert checks the correctness of the statement itself too before selecting it

- [x] a. Reduce the weight decay factor as you might have regularized too much the network ✔
- [x] b. Add skip connections to be sure vanishing gradient is not preventing layers close to the input to be updated ✔
- [x] c. Look at the distribution of the norms of the gradients, from that you should be able to tell if some exploding or vanishing gradient is ongoing ✔
- [ ] d. Replace sigmoidal activation unit with hyperbolic tangent ones in the hidden layers to rule out any possible ongoing vanishing gradient due to activation functions
- [x] e. Add more neurons as you might have not sized the model to be powerful enough and you are possibly underfitting ✔
- [ ] f. Introduce gradient clipping, i.e., saturate the value of gradient updates, as there might be clearly some exploding gradient ongoing and there is no means to realize it beside trial and error
- [ ] g. Double the learning rate; this should fix a possible ongoing vanishing gradient

Your answer is correct.

Le risposte corrette sono:
Add more neurons as you might have not sized the model to be powerful enough and you are possibly underfitting,

Reduce the weight decay factor as you might have regularized too much the network,

Add skip connections to be sure vanishing gradient is not preventing layers close to the input to be updated,

Look at the distribution of the norms of the gradients, from that you should be able to tell if some exploding or vanishing gradient is ongoing

Domanda 3

Completo

Punteggio ottenuto 3,00 su 5,00

With reference to the Neural Turing Machines answer the following questions (use the letters to organize your answer)

a) Describe the Neural Turing Machine idea and how read / write instructions are implemented
b) Describe the architecture of a Neural Turing Machine and the attention mechanisms in it
c) Provide 2 examples of tasks that could be suited for a Neural Turing Machine better than a seq2seq model and motivate your answer

A classical Turing machine operates on the idea that there is a long stream of cells in which the NTM can perform operations, namly read and write to, rolling this “tape” back and forth as needed. A NTM applies this idea to neural networks, making available the possibility to write and read from a bank memory outside the neural network.

Attention is a powerful way for, given a set of inputs, obtain a vector of weights that somehow give importance to more inputs rather than others. With this in mind, we can emply attention in many types of networks improving their performance by focusing more on the relevant inputs to provide the outputs.

In NTMs the attention mechanisms works by giving suitable attention weights to the cells and allowing the neural network to focus on the most important ones for performing operations.

In seq2seq model the idea is that we have an encoder and a decoder, the encoder is responsible for taking an input sequence and encode it a lower dimensional vector, called context vector, which is then fed to the decoder for producing the desired output. Possible tasks for seq2seq are machine translation or image captioning.

A NTM might be better suited when the information at our disposal is too large to fit in the seq2seq model and be learnt, thus employing the idea of writing to an external memory outlifts some complexity from the process.

Commento:
a) 1/2 no read write described

b) 1/2

c) 1/1

Domanda **4**

Parzialmente corretta

Punteggio ottenuto 0,50 su 1,00

Assume you are designing a word embedding for your custom text dataset on very specialized documents, so you want to train from scratch a word2vec model (continuous bag of words) like the one in the picture to extract an embedding matrix from terms to vectors.

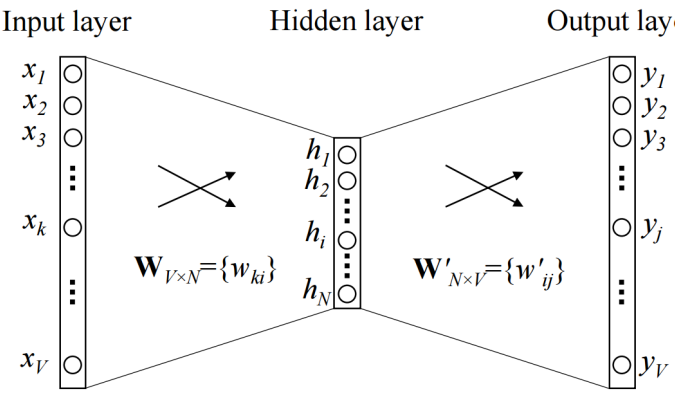


Figure 1: A simple CBOW model with only one word in the context

The dataset you are working with has 1380 unique terms and you want to embed them into vectors of 32 elements. Taking into account also the picture, insert the missing numbers in the following by considering that W' is the transpose of W. You can only input digits {0, 1, ..., 9} with no spaces or formulas in the following.

The number of parameters in the resulting embedding matrix which goes from one hot encoded term to their encoding which will be extracted out of the network is

44160

✔ and the overall number of the trainable parameters of the network at training time when the CBOW is trained is

44160

✖ .

Domanda **5**

Parzialmente corretta

Punteggio ottenuto 1,00 su 2,00

Let us consider the object detection problem. Mark all the sentences that are correct.

Please, refrain from marking answers that are "technically possible in Python", and stick to those answers that a Neural Network expert (as you are expected to be) would give and the solution he would adopt.

Scegli una o più alternative:

- [ ] a. If you want to train an R-CNN to detect L different types of objects, you need to train a classifier using L classes, as the objectiveness score will take care of excluding detections on the background.
- [ ] b. There are many object detection networks, but only those attempting to segment objects have skip connections.
- [ ] c. YOLO is not an object detection network. It is primarily used in wheather forecasting and to detect tropical cyclons
- [ ] d. It is in principle possible to implement an object detection network mimicking the architecture of Faster R-CNN to classify heartbeats in ECG tracing, which is 1D signal.
- [x] e. In Faster R-CNN the anchors identify the regions that are then fed to a GAP to extract a vector that goes to the output heads ✖ FALSE: the anchors are used to identify labels for objectiveness score, there is no GAP involved in Faster R-CNN.
- [ ] f. The output of Faster R-CNN is actually the output of the Region Proposal Network (RPN), which contains bounding boxes and their class estimates.
- [x] g. Object detection networks can return a varying number of outputs for a single input image. ✔ TRUE, depending on the content of the image
- [ ] h. When performing object / event detection in a time series, it is possible to adopt a MLP taking as input the time series.
- [x] i. Some object detection network leverage a hand-crafted, deterministic algorithm to identify regions that potentially contain objects. ✔ TRUE: R-CNN is done in this way.

Your answer is partially correct.

Hai selezionato correttamente 2.
Le risposte corrette sono:
Some object detection network leverage a hand-crafted, deterministic algorithm to identify regions that potentially contain objects.,

It is in principle possible to implement an object detection network mimicking the architecture of Faster R-CNN to classify heartbeats in ECG tracing, which is 1D signal.,

Object detection networks can return a varying number of outputs for a single input image.

Domanda **6**

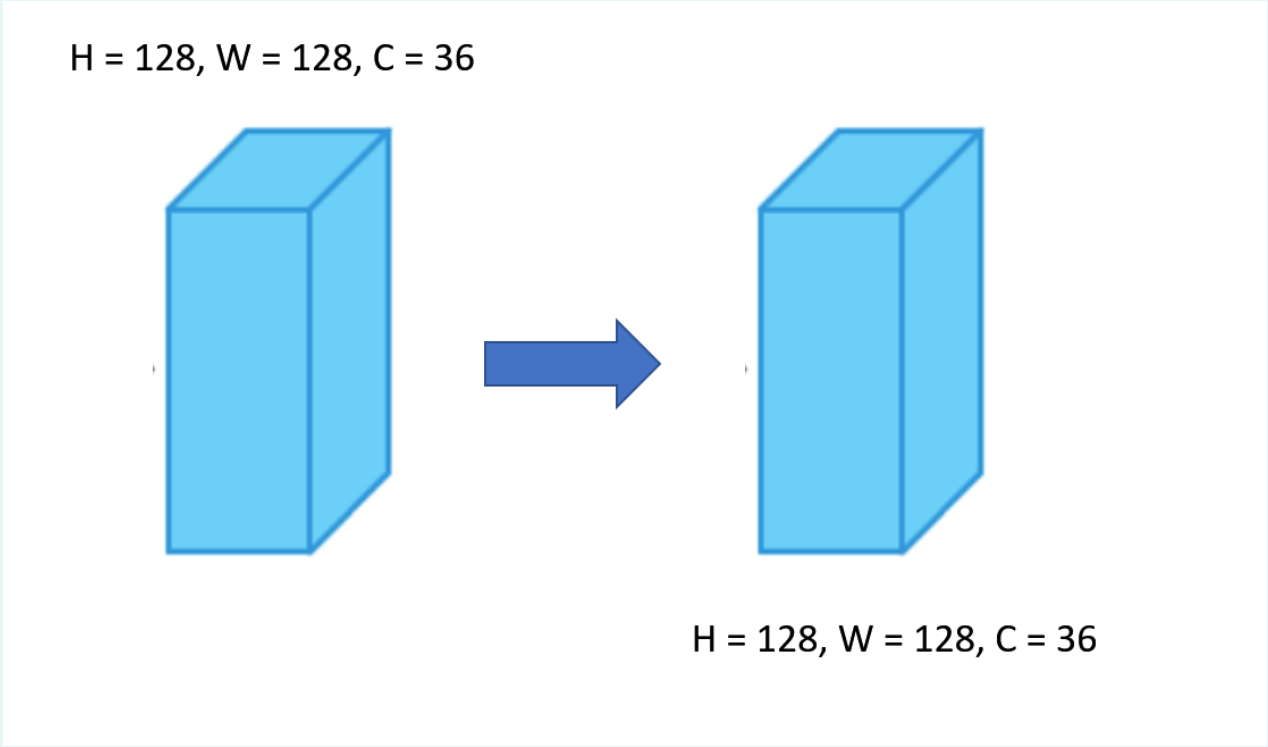
Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Which of the following layers can be technically plugged in between the activations maps in the following scheme

Lines of code refer to the following import.

import tensorflow as tf
tfkl = tf.keras.layers



Scegli una o più alternative:

- [x] a. tfkl.Conv2D(36, kernel_size=3, padding='same', activation='relu') ✔ YES!
- [x] b. `layers.Activation(activations.relu)` ✔ YES: activations do not change sizes
- [ ] c. GlobalAveragingPooling2D()
- [ ] d. `tfkl.UpSampling2D()`
- [x] e. tfkl.MaxPooling2D(x, `pool_size=(3, 3), strides = (1,1), padding = 'same'`) ✔ YES: this does not reduce the spatial extent, as the strides is (1,1)
- [ ] f. it is not possible to say, since we don't know what will happen in the next layers
- [ ] g. tfkl.Conv2D(64, kernel_size=1, padding='same', activation='relu')

Your answer is correct.

Le risposte corrette sono:

tfkl.Conv2D(36, kernel_size=3, padding='same', activation='relu')
,

tfkl.MaxPooling2D(x, pool_size=(3, 3), strides = (1,1), padding = 'same')
,

```
layers.Activation(activations.relu)
```

Domanda 7

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

The following questions concern Autoencoders. Select all those sentences that are true.

As in the previous section, act as a neural network expert, do not consider true what is simply possible to program.

Scegli una o più alternative:

- [x] a. Autoencoders always need to have a bottleneck for the latent representation. ✔ TRUE: otherwise they might have the potential to learn the identify function
- [x] b. Convolutional autoencoders architectures can be trained to restore degraded images. ✔ TRUE: we can just synthetically degrade images (e.g. by adding noise) to get training
- [ ] c. Plain vanilla autoencoders can generate high-quality images of human faces.
- [x] d. Autoencoders can be used to pre-train the backbone of a classifier, when there are little annotated samples available. ✔ TRUE: as we have explained during lectures
- [ ] e. Convolutional autoencoders are made of convolutional, activations and pooling layers. It is impossible to have Dense layers within convolutional autoencoders.
- [ ] f. Autoencoders are made of two networks, the Generator and the Discriminator, which are randomly initialized. At the end of training we keep only the Generator.
- [ ] g. Autoencoders are special types of neural network that takes as input an image and returns a vector.
- [ ] h. Autoencoders are ultra-fast model that can be used in high trouput datastreams to isolate patterns coforming to known shapes.
- [ ] i. Like most neural networks, autoencoders require annotated input for training.

Your answer is correct.

Le risposte corrette sono:
Convolutional autoencoders architectures can be trained to restore degraded images.,

Autoencoders can be used to pre-train the backbone of a classifier, when there are little annotated samples available.,

Autoencoders always need to have a bottleneck for the latent representation.

Domanda 8

Risposta non data

Non valutata

Consider the following code

```
import tensorflow as tf
tfkl = tf.keras.layers
tfk = tf.keras

def get_model_2(input_shape, features):

    # Output Layer
    input_layer = tfkl.Input(shape=input_shape, name='input_layer')

    # Block 1
    c1 = tfkl.Conv2D(64, kernel_size=5, padding='same', activation='relu',
name='block_1_conv_1')(input_layer)
    c1 = tfkl.MaxPooling2D(name='block_1_mp')(c1)
    c1 = tfkl.Dropout(0.2, name='block_1_dropout')(c1)
    s1 = tfkl.Conv2D(128, kernel_size=5, padding='same', activation='relu',
name='block_1_conv_2')(c1)
    s1 = tfkl.Conv2D(64, kernel_size=5, padding='same', activation='relu',
name='block_1_conv_3')(s1)
    s1 = tfkl.Add(name='block_1_add')([c1,s1])

    # Block 2
    c2 = tfkl.Conv2D(128, kernel_size=5, padding='same', activation='relu',
name='block_2_conv_1')(s1)
    c2 = tfkl.MaxPooling2D(name='block_2_mp')(c2)
    c2 = tfkl.Dropout(0.2, name='block_2_dropout')(c2)
    s2 = tfkl.Conv2D(256, kernel_size=5, padding='same', activation='relu',
name='block_2_conv_2')(c2)
    s2 = tfkl.Conv2D(128, kernel_size=5, padding='same', activation='relu',
name='block_2_conv_3')(s2)
    s2 = tfkl.Add(name='block_2_add')([c2,s2])
    s2 = tfkl.GlobalAveragePooling2D(name='block_2_gap')(s2)

    # Output Layer
    output_layer = tfkl.Dense(features, activation="linear", name='output_layer')(s2)

    model = tf.keras.Model(inputs=input_layer, outputs=output_layer)
    model.compile(
        loss = tfk.losses.MeanSquaredError(),
        optimizer = tfk.optimizers.Adam(),
        metrics = [
            tfk.metrics.MeanAbsoluteError(),
            tfk.metrics.MeanAbsolutePercentageError()
            ]
        )
    return model

input_shape = (32,32,3)
features = 6
model = get_model_2(input_shape=input_shape, features=features)
model.summary()
tfk.utils.plot_model(model)
```

The last command returns the model summary. Below you can find a reference for you to work on the model

Please fill in the scheme and upload it. This is a backup option, the following question will be assessed

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_layer (InputLayer)
```

```
 block_1_conv_1 (Conv2D)
 block_1_mp (MaxPooling2D)
 block_1_dropout (Dropout)
 block_1_conv_2 (Conv2D)
 block_1_conv_3 (Conv2D)
 block_1_add (Add)
 block_2_conv_1 (Conv2D)
 block_2_mp (MaxPooling2D)
 block_2_dropout (Dropout)
 block_2_conv_2 (Conv2D)
 block_2_conv_3 (Conv2D)
 block_2_add (Add)
 block_2_gap (GAP2D)
 output_layer (Dense)
_________________________________________________________________
Total params:
Trainable params:
Non-trainable params:
_________________________________________________________________
```

```
Model: "model"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_layer (InputLayer)       [(None, 32, 32, 3)]  0           []
 block_1_conv_1 (Conv2D)        (None, 32, 32, 64)   4864        ['input_layer[0][0]']
 block_1_mp (MaxPooling2D)      (None, 16, 16, 64)   0           ['block_1_conv_1[0][0]']
 block_1_dropout (Dropout)      (None, 16, 16, 64)   0           ['block_1_mp[0][0]']
 block_1_conv_2 (Conv2D)        (None, 16, 16, 128)  204928      ['block_1_dropout[0][0]']
 block_1_conv_3 (Conv2D)        (None, 16, 16, 64)   204864      ['block_1_conv_2[0][0]']
 block_1_add (Add)              (None, 16, 16, 64)   0           ['block_1_dropout[0][0]',
                                                                  'block_1_conv_3[0][0]']
 block_2_conv_1 (Conv2D)        (None, 16, 16, 128)  204928      ['block_1_add[0][0]']
 block_2_mp (MaxPooling2D)      (None, 8, 8, 128)    0           ['block_2_conv_1[0][0]']
 block_2_dropout (Dropout)      (None, 8, 8, 128)    0           ['block_2_mp[0][0]']
 block_2_conv_2 (Conv2D)        (None, 8, 8, 256)    819456      ['block_2_dropout[0][0]']
 block_2_conv_3 (Conv2D)        (None, 8, 8, 128)    819328      ['block_2_conv_2[0][0]']
 block_2_add (Add)              (None, 8, 8, 128)    0           ['block_2_dropout[0][0]',
                                                                  'block_2_conv_3[0][0]']
 block_2_gap (GlobalAveragePool  (None, 128)         0           ['block_2_add[0][0]']
 output_layer (Dense)           (None, 6)            774         ['block_2_gap[0][0]']
==================================================================================================
Total params: 2,259,142
Trainable params: 2,259,142
Non-trainable params: 0
__________________________________________________________________________________________________
```

Domanda 9

Parzialmente corretta

Punteggio ottenuto 5,00 su 5,00

Remember that this question will be automatically rated, therefore you have to consider the following details very carefully:

- when the activations become a vector, these are stacked along the third dimension (the channel, C).
  Therefore, if your latent representation becomes a 3 element vector, enter as sizes H:1 W:1 C:3.
- enter digits **only** and do not user "." or "," or "–"" or "[]". Say write "1234" and not "1,234" or "1.234". Write "0", not "–" or "[]".
- do not enter formulas, just resulting numbers
- connections, if any have been omitted in this summary

All the numbers have to be entered here, the text in the previous question is just meant for you to design the solution.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_layer (InputLayer)     H:
```
32
✔ , W:
32
✔ , C:
3
✔ Params #:
0
✔
```
block_1_conv_1 (Conv2D)      H:
```
32
✔ , W:
32
✔ , C:
64
✔ Params #:
4864
✔
```
block_1_mp (MaxPooling2D)    H:
```
16
✔ , W:
16
✔ , C:
64
✔ Params #:
0
✔
```
block_1_dropout (Dropout)    H:
```
16
✔ , W:
16
✔ , C:
64
✔ Params #:
0
✔
```
block_1_conv_2 (Conv2D)      H:
```
16
✔ , W:
16
✔ , C:
128
✔ Params #:
204928
✔
```
```

block_1_conv_3 (Conv2D) H:
16
✔ , W:
16
✔ , C:
64
✔ Params #:
204864
✔
block_1_add (Add) H:
16
✔ , W:
16
✔ , C:
64
✔ Params #:
0
✔
block_2_conv_1 (Conv2D) H:
16
✔ , W:
16
✔ , C:
128
✔ Params #:
204928
✔
block_2_mp (MaxPooling2D) H:
8
✔ , W:
8
✔ , C:
128
✔ Params #:
0
✔
block_2_dropout (Dropout) H:
8
✔ , W:
8
✔ , C:
128
✔ Params #:
0
✔
block_2_conv_2 (Conv2D) H:
8

✔ , W:
8
✔ , C:
256
✔ Params #:
819456
✔
block_2_conv_3 (Conv2D) H:
8
✔ , W:
8
✔ , C:
128
✔ Params #:
819328
✔
block_2_add (Add) H:
8
✔ , W:
8
✔ , C:
128
✔ Params #:
0
✔
block_2_gap (GAP2D) H:
1
✔ , W:
1
✔ , C:
128
✔ Params #:
0
✔
output_layer (Dense) H:
1
✔ , W:
1
✔ , C:
6
✔ Params #:
774
✔
==========
Total params:
2259142
✔

Trainable params:

2259142

✔

Non-trainable params:

0

✔

Layer (type) Output Shape Param # Connected to

==========

input_layer (InputLayer) [(None, 32, 32, 3)] 0 []

block_1_conv_1 (Conv2D) (None, 32, 32, 64) 4864 ['input_layer[0][0]']

block_1_mp (MaxPooling2D) (None, 16, 16, 64) 0 ['block_1_conv_1[0][0]']

block_1_dropout (Dropout) (None, 16, 16, 64) 0 ['block_1_mp[0][0]']

block_1_conv_2 (Conv2D) (None, 16, 16, 128) 204928 ['block_1_dropout[0][0]']

block_1_conv_3 (Conv2D) (None, 16, 16, 64) 204864 ['block_1_conv_2[0][0]']

block_1_add (Add) (None, 16, 16, 64) 0 ['block_1_dropout[0][0]',

'block_1_conv_3[0][0]']

block_2_conv_1 (Conv2D) (None, 16, 16, 128) 204928 ['block_1_add[0][0]']

block_2_mp (MaxPooling2D) (None, 8, 8, 128) 0 ['block_2_conv_1[0][0]']

block_2_dropout (Dropout) (None, 8, 8, 128) 0 ['block_2_mp[0][0]']

block_2_conv_2 (Conv2D) (None, 8, 8, 256) 819456 ['block_2_dropout[0][0]']

block_2_conv_3 (Conv2D) (None, 8, 8, 128) 819328 ['block_2_conv_2[0][0]']

block_2_add (Add) (None, 8, 8, 128) 0 ['block_2_dropout[0][0]',

'block_2_conv_3[0][0]']

block_2_gap (GlobalAveragePool (None, 128) 0 ['block_2_add[0][0]']

output_layer (Dense) (None, 6) 774 ['block_2_gap[0][0]']

==========

Total params: 2,259,142

Trainable params: 2,259,142

Non-trainable params: 0

Grading Criteria

Each answers is assessed but not all correct corresponds to points (some are too simple, some are repeated).

Some wrong answers contributes negatively to your final grade, in particular those with very apparent errors (e.g. setting parameters to a gap or a relu)

Thefore, the final grade can be in principle negative.

Domanda **10**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Mark all the tasks that the above network can be trained for.

Please, refrain from marking answers that are "technically and eventually possible in Python", and stick to those tasks that a Neural Network expert (as you are expected to be) would use this network for.

Needless to say, assume you are always provided with the required training data for these tasks, while you cannot change the architecture and training options for the above model.

Scegli una o più alternative:

- [ ] a. Localizing the centers of eyes, of nostrils and ears in the picture of the driving licence.
- [ ] b. Classify the the digits from 0 to 5 in from a subset of MNIST dataset.
- [x] c. Counting the number of cats, of dogs, of traffic lights, of bycicles, of cars and of persons in a picture taken from "Street View" ✔ True, that's a regression problem
- [ ] d. Detecting persons and cats, and for each of these return a bounding box
- [ ] e. Given a picture of a human face, drawing a bounding box around the face and say whether the person is a male or female
- [x] f. Given a picture of a car, estimate its weight, its speed, and localize it through a bounding box. ✔ This is a six-output regression problem.
- [x] g. Given the picture of a human face, estimate the tip of ears and the tip of nose. ✔ TRUE: This is a regression problem over 6 image coordinates

Your answer is correct.

Le risposte corrette sono:
Counting the number of cats, of dogs, of traffic lights, of bycicles, of cars and of persons in a picture taken from "Street View" ,

Given a picture of a car, estimate its weight, its speed, and localize it through a bounding box.,

Given the picture of a human face, estimate the tip of ears and the tip of nose.