

Artificial Neural Networks and Deep Learning

Keras tutorial - 14/10/2020

Francesco Lattari, PhD student (francesco.lattari@polimi.it)

Artificial Intelligence and Robotics Laboratory



POLITECNICO
MILANO 1863

Class `tf.keras.Model`

https://www.tensorflow.org/api_docs/python/tf/keras/Model

`tf.keras.callbacks`

https://www.tensorflow.org/api_docs/python/tf/keras/callbacks

- Save model
- Learning visualization
- ...

2. Training model

```
model.fit(  
    x=None,  
    y=None,  
    batch_size=None,  
    epochs=1,  
    verbose=1,  
    callbacks=None,  
    validation_split=0.0,  
    validation_data=None,  
    shuffle=True,  
    class_weight=None,  
    sample_weight=None,  
    initial_epoch=0,  
    steps_per_epoch=None,  
    validation_steps=None,  
    validation_freq=1,  
    max_queue_size=10,  
    workers=1,  
    use_multiprocessing=False,  
    **kwargs)
```

Class `tf.keras.Model`

https://www.tensorflow.org/api_docs/python/tf/keras/Model

- Compute loss and metrics in test mode

```
evaluate(
    x=None,
    y=None,
    batch_size=None,
    verbose=1,
    sample_weight=None,
    steps=None,
    callbacks=None,
    max_queue_size=10,
    workers=1,
    use_multiprocessing=False
)
```

Return:

test loss or list of scalars (for multiple outputs and metrics)

- Compute model output:

```
predict(
    x,
    batch_size=None,
    verbose=0,
    steps=None,
    callbacks=None,
    max_queue_size=10,
    workers=1,
    use_multiprocessing=False
)
```

Return:

predictions as tensors

Class `tf.keras.callbacks.ModelCheckpoint`

https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/ModelCheckpoint

- Save model during learning

```
ModelCheckpoint(  
    filepath,  
    monitor='val_loss',  
    verbose=0,  
    save_best_only=False,  
    save_weights_only=False,  
    mode='auto',  
    save_freq='epoch',  
    **kwargs  
)
```

Class `tf.keras.callbacks.ModelCheckpoint`

https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/ModelCheckpoint

- Save model during learning

```
ModelCheckpoint(  
    filepath,  
    monitor='val_loss',  
    verbose=0,  
    save_best_only=False,  
    save_weights_only=False,  
    mode='auto',  
    save_freq='epoch',  
    **kwargs  
)
```

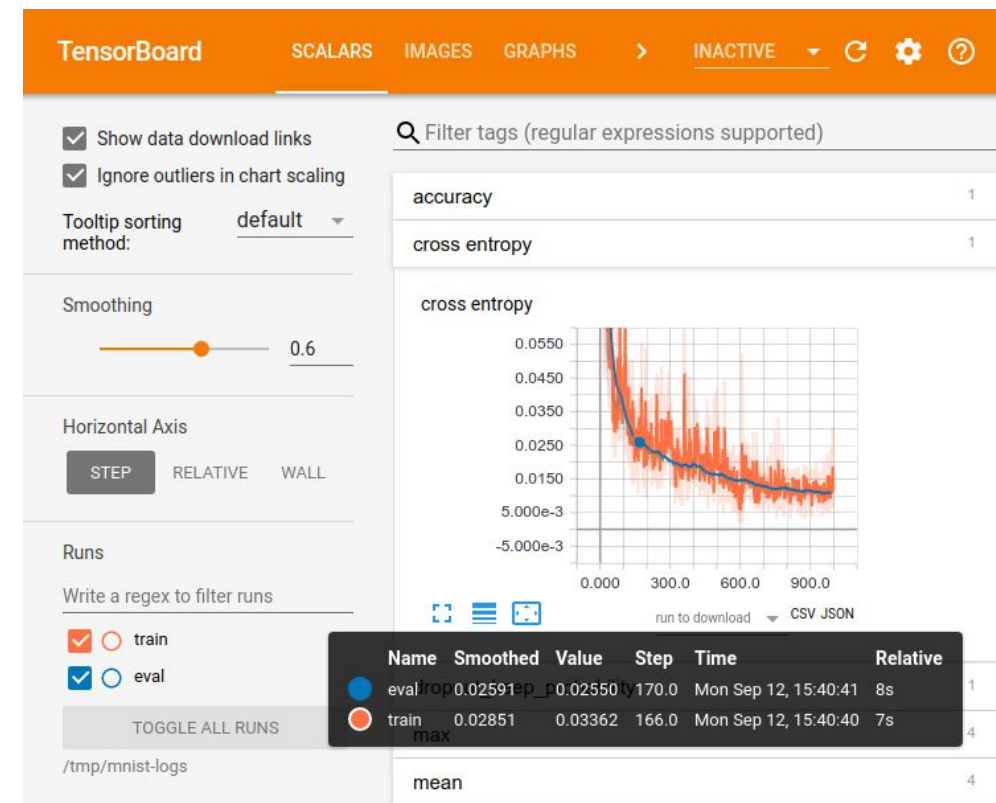
True: save only model weights

- Save
`model.save_weights('checkpoint_path/filename.ckpt')`
- Restore.
`model = Model(...)`
`model.load_weights('checkpoint_path/filename.ckpt')`

False: save the entire model

- Save
`model.save('model_path/filename')`
- Restore.
`model = tf.keras.models.load_model('model_path/filename')`

TensorBoard: Tensorflow's visualization toolkit



- Plotting scalars
 - e.g., losses, accuracy, gradients, etc.
- Show images
 - e.g., layer activations, segmentation results, filters, etc.
- Plotting histograms
 - e.g., gradients and weights distribution
- Show the model graph
-

Class `tf.keras.callbacks.Tensorboard`

https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/TensorBoard

```
Tensorboard(  
    log_dir='logs',  
    histogram_freq=0,  
    write_graph=True,  
    write_images=False,  
    update_freq='epoch',  
    profile_batch=2,  
    embeddings_freq=0,  
    embeddings_metadata=None,  
    **kwargs  
)
```

How to open Tensorboard in browser:

1. Run from terminal: *tensorboard --logdir /path/to/exps --port PORT*
2. Open *127.0.0.1:PORT* in your browser

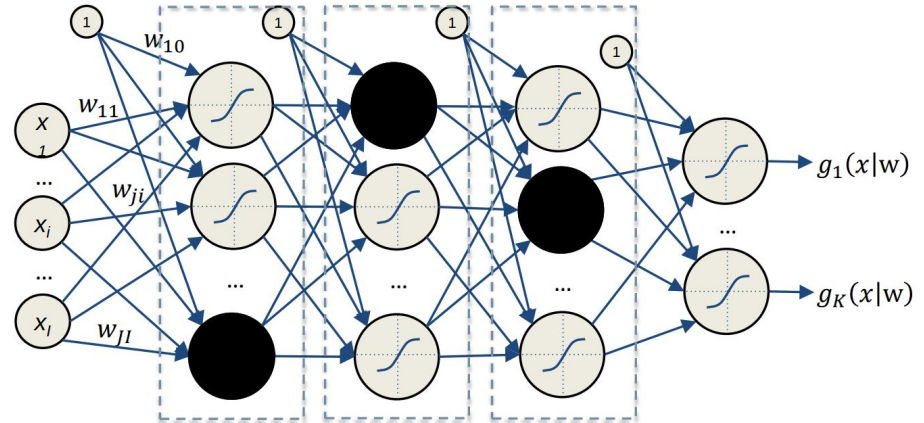
- Reduce number of parameters
- Dropout
- Weight decay
- Early stopping

- Reduce number of parameters
- Dropout
- Weight decay
- Early stopping

Class `tf.keras.layers.Dropout`

https://www.tensorflow.org/api_docs/python/tf/keras/layers/Dropout

Each hidden unit is turned off with probability equal to 'rate' (main parameter)



- Reduce number of parameters
- Dropout
- Weight decay
- Early stopping

Class `tf.keras.regularizers.l2`

https://www.tensorflow.org/api_docs/python/tf/keras/regularizers/l2

$$\underset{w}{\operatorname{argmin}} \underbrace{\sum_{n=1}^N (t_n - g(x_n|w))^2}_{\text{Fitting}} + \underbrace{\gamma \sum_{q=1}^Q (w_q)^2}_{\text{Regularization}}$$

- e.g., fully-connected layer

```
tf.keras.layers.Dense(  
    units,  
    activation=None,  
    use_bias=True,  
    kernel_initializer='glorot_uniform',  
    bias_initializer='zeros',  
    kernel_regularizer=tf.keras.regularizers.l2(0.001),  
    bias_regularizer=None,  
    activity_regularizer=None,  
    kernel_constraint=None,  
    bias_constraint=None,  
    **kwargs,  
)
```

- Reduce number of parameters
- Dropout
- Weight decay
- Early stopping

Class `tf.keras.callbacks.EarlyStopping`

https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/EarlyStopping

```
EarlyStopping(  
    monitor='val_loss',  
    min_delta=0,  
    patience=0,  
    verbose=0,  
    mode='auto',  
    baseline=None,  
    restore_best_weights=False  
)
```