



PUC Minas

# Frameworks de Deep Learning

Professor: Renan Santos Mendes

Keras



# Callbacks

- Callbacks são objetos do Keras que podem ser passados para o método fit durante o treinamento de um modelo.
- Permitem que você **especifique ações** a serem executadas em determinados momentos durante o treinamento, como no início ou no final de uma época.
- **Permitem monitorar métricas** de treinamento em tempo real e tomar decisões com base nesses dados.
- Permitem ajustar hiperparâmetros durante o treinamento, como a taxa de aprendizagem ou o tamanho do lote.



# Callbacks

- Permitem salvar o modelo ou visualizar o progresso do treinamento em um tensorboard.
- São especialmente úteis para evitar o overfitting durante o treinamento, por exemplo, parando o treinamento prematuramente quando a perda de validação começa a aumentar.
- O Keras possui vários callbacks integrados, como o **ModelCheckpoint** para salvar o modelo e o **EarlyStopping** para interromper o treinamento prematuramente quando a perda de validação começa a aumentar.
- Você também pode criar seus próprios callbacks personalizados para executar ações específicas durante o treinamento.





# Callbacks

```
from keras.callbacks import EarlyStopping

early_stopping = EarlyStopping(monitor='val_loss',
                               patience=1,
                               restore_best_weights=True,
                               verbose=1)

model = get_compiled_model()
model.fit(X_train, y_train,
          epochs=100,
          batch_size=32,
          validation_split=0.2,
          callbacks=[early_stopping])
```

# Uso de Strings X Objetos





# Uso de Strings X Objetos

- Em diversas ocasiões no Keras podemos utilizar strings ou criar objetos;
- Ex.:
  - Funções de ativação
  - Otimizadores
  - Métricas
- Em geral, quando usamos strings, o elemento criado é pouco customizável;
- Quando criamos o elemento via POO, torna-se altamente customizável

# Uso de Strings X Objetos

```
opt = 'SGD'
opt = 'RMSprop'
opt = 'Adam'

model.compile(loss='categorical_crossentropy',
              optimizer=opt,
              metrics=['accuracy'])
```



```
lr_schedule = keras.optimizers.schedules.ExponentialDecay(
    initial_learning_rate=1e-2,
    decay_steps=100,
    decay_rate=0.9)

optimizer = keras.optimizers.SGD(learning_rate=lr_schedule)

model.compile(loss='categorical_crossentropy',
              optimizer=optimizer,
              metrics=['accuracy'])
```

# Técnicas para Corrigir Overfitting







# Correção dos Problemas

Existem algumas formas de contornar o problema de overfitting:

- Early stopping;
- Reduzir complexidade do modelo;
- Regularização;
- Dropout;
- Data augmentation.