

Machine Learning Aplicado

Framework Keras

Ingeniería Civil Informática
Escuela de Ingeniería Informática
Universidad de Valparaíso

Profesor: Aaron Ponce Sandoval
correo: aaron.ponce@uv.cl

- ❑ Keras es un marco de trabajo de aprendizaje profundo de alto nivel
- ❑ Proporciona una API fácil de usar para construir y entrenar modelos de redes neuronales.



Keras Framework

- ❑ Keras es una API de aprendizaje profundo escrita en Python
- ❑ Actualmente se ejecuta sobre la plataforma de aprendizaje automático TensorFlow.
- ❑ Fue desarrollado para permitir la experimentación rápida.
- ❑ Se basa en la idea de obtener resultados lo más rápido posible.



DEEP LEARNING
con Python

François Chollet

ANAYA

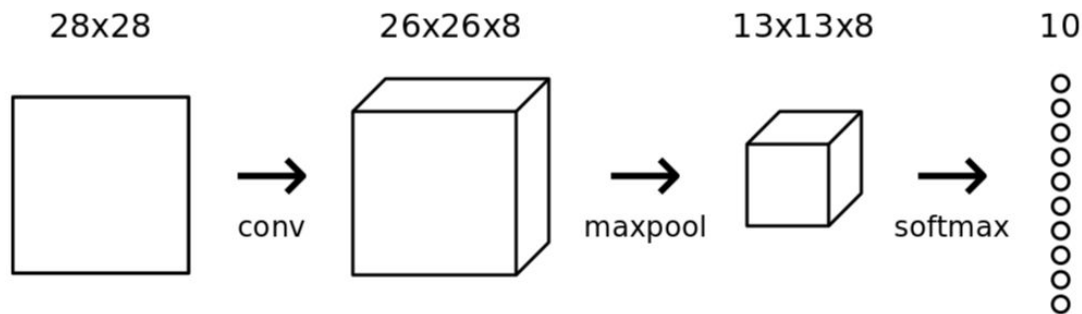


Simple: pero no simplista. Keras reduce la carga cognitiva del desarrollador para que pueda concentrarse en las partes del problema que realmente importan.

Flexible: Keras adopta el principio de revelación progresiva de la complejidad: los flujos de trabajo simples deben ser rápidos y fáciles, mientras que los flujos de trabajo arbitrariamente avanzados deben ser posibles a través de un camino claro que se basa en lo que ya ha aprendido

Potente: Keras proporciona un rendimiento y una escalabilidad sólida en la industria: lo utilizan organizaciones y empresas, incluidas la NASA, YouTube o Waymo.

- ❑ Capas: las capas son los bloques de construcción fundamentales de una red neuronal en Keras.
- ❑ Las capas son como "ladrillos" que se apilan para construir una arquitectura de red neuronal.
- ❑ Keras proporciona una variedad de capas, como capas de entrada, capas densas, capas convolucionales, capas de agrupamiento, capas de normalización, capas de activación y muchas otras.



```
from tensorflow.keras import layers  
layer = layers.Dense(32, activation='relu')  
inputs = tf.random.uniform(shape=(10, 20))  
outputs = layer(inputs)
```

- ❑ Los modelos en Keras son contenedores que se utilizan para organizar capas en una arquitectura de red neuronal.
- ❑ Los modelos se pueden crear usando la API secuencial de Keras, que es una forma sencilla de construir modelos en los que las capas se apilan una tras otra, o utilizando la API funcional de Keras, que permite construir modelos más complejos y flexibles.

```
# Modelo I
model = tf.keras.Sequential()
model.add(tf.keras.layers.Dense(8, input_shape=(16,)))
model.add(tf.keras.layers.Dense(4))
# Modelo II
model = tf.keras.Sequential()
model.add(tf.keras.Input(shape=(16,)))
model.add(tf.keras.layers.Dense(8))
```


- ❑ Las funciones de activación son funciones matemáticas que se aplican a las salidas de las capas para introducir no linealidad en la red neuronal.
- ❑ Keras proporciona una variedad de funciones de activación, como la función sigmoide, la función ReLU, la función softmax y muchas otras.

Funcion de Activacion

```
from tensorflow.keras import layers  
  
from tensorflow.keras import activations  
  
model.add(layers.Dense(64))  
  
model.add(layers.Activation(activations.relu)) # forma 1
```

```
model.add(layers.Dense(64, activation='relu')) # forma 2
```

- ❑ Los optimizadores son algoritmos que se utilizan para ajustar los pesos de las conexiones de la red neuronal durante el entrenamiento para minimizar la función de pérdida.
- ❑ Keras proporciona una variedad de optimizadores, como el descenso de gradiente estocástico (SGD), el Adam y muchos otros.

Optimizador Adam

```
from tensorflow import keras
from tensorflow.keras import layers
model = keras.Sequential()
model.add(layers.Dense(64, kernel_initializer='uniform', input_shape=(10,)))
model.add(layers.Activation('softmax'))
opt = keras.optimizers.Adam(learning_rate=0.01)#optimizador Adam
model.compile(loss='categorical_crossentropy', optimizer=opt)
```

- ❑ Las funciones de pérdida son funciones matemáticas que se utilizan para medir la discrepancia entre la salida de la red neuronal y la salida esperada.
- ❑ Keras proporciona una variedad de funciones de pérdida, como el error cuadrático medio (MSE), la entropía cruzada y muchas otras.

Loss Function

```
from tensorflow import keras
from tensorflow.keras import layers
model = keras.Sequential()
model.add(layers.Dense(64, kernel_initializer='uniform', input_shape=(10,)))
model.add(layers.Activation('softmax'))
opt = keras.optimizers.Adam(learning_rate=0.01)#optimizador Adam
model.compile(loss='categorical_crossentropy', optimizer=opt)# funcion de
perdida
```

- ❑ Las métricas son funciones que se utilizan para evaluar el rendimiento de la red neuronal durante el entrenamiento y la evaluación.
- ❑ Keras proporciona una variedad de métricas, como la precisión, la tasa de falsos positivos y muchas otras.

```
model.compile(  
    optimizer='adam',  
    loss='mean_squared_error',  
    metrics=[  
        metrics.MeanSquaredError(),  
        metrics.AUC(),  
    ]  
)
```


- ❑ Los callbacks son funciones que se utilizan para personalizar el comportamiento de Keras durante el entrenamiento de la red neuronal.
- ❑ Los callbacks se pueden utilizar, por ejemplo, para guardar el modelo después de cada época, para ajustar la tasa de aprendizaje durante el entrenamiento o para detener el entrenamiento si el modelo deja de mejorar.

Callbacks

```
my_callbacks = [  
    tf.keras.callbacks.EarlyStopping(patience=2),  
  
    tf.keras.callbacks.ModelCheckpoint(filepath='model.{epoch:02d}-{val_loss:.2f}.h5'),  
  
    tf.keras.callbacks.TensorBoard(log_dir='./logs'),  
]  
  
model.fit(dataset, epochs=10, callbacks=my_callbacks)
```