
Keras Python Library

Επιμέλεια: Α.Δρακόπουλος

Πίνακας περιεχομένων

Τι είναι το Keras;	3
Χαρακτηριστικά του Keras	3
Κύριες Συναρτήσεις και Πακέτα του Keras	3
Δημιουργία και Εκπαίδευση Νευρωνικού Δικτύου	3
1. Διαδοχικό Μοντέλο (Sequential Model)	3
2. Λειτουργικό API (Functional API)	4
3. Σύνοψη Μοντέλου: model.summary()	4
Προχωρημένες Δυνατότητες του Keras	4
4. Callbacks	4
5. Αποθήκευση και Φόρτωση Μοντέλου	5
Αποθήκευση	5
Φόρτωση	5
6. Προεπεξεργασία Δεδομένων	5
7. Αξιολόγηση Μοντέλου: model.evaluate()	5
8. Πρόβλεψη: model.predict()	5
Υποστηριζόμενα Layers στο Keras	5
Συμπέρασμα	6

Τι είναι το Keras;

Το **Keras** είναι μια υψηλού επιπέδου βιβλιοθήκη για τη δημιουργία και την εκπαίδευση νευρωνικών δικτύων, γραμμένη σε Python. Είναι ενσωματωμένη στο **TensorFlow** από την έκδοση 2.0 και παρέχει έναν εύκολο και διαισθητικό τρόπο για την ανάπτυξη μοντέλων βαθιάς μάθησης (Deep Learning). Το Keras υποστηρίζει γρήγορη πρωτοτυποποίηση, γεγονός που το καθιστά ιδανικό για την έρευνα και την παραγωγή.

Χαρακτηριστικά του Keras

1. **Υψηλού επιπέδου API:** Διαισθητική διεπαφή για τη δημιουργία νευρωνικών δικτύων.
2. **Ευελιξία:** Υποστηρίζει πλήθος backends (π.χ., TensorFlow, Theano, CNTK).
3. **Υποστήριξη CPU/GPU:** Επιτρέπει την ταχύτατη εκπαίδευση χρησιμοποιώντας GPU.
4. **Πολυμορφία μοντέλων:** Υποστηρίζει τόσο διαδοχικά (sequential) όσο και λειτουργικά (functional) μοντέλα.
5. **Προσαρμοσμένα layers και loss functions:** Δυνατότητα δημιουργίας προσαρμοσμένων αρχιτεκτονικών.

Κύριες Συναρτήσεις και Πακέτα του Keras

1. `keras.models`: Για τη δημιουργία και την εκπαίδευση μοντέλων.
2. `keras.layers`: Για την προσθήκη διαφορετικών τύπων επιπέδων (layers).
3. `keras.optimizers`: Για τη βελτιστοποίηση της εκπαίδευσης.
4. `keras.losses`: Για τον ορισμό της συνάρτησης κόστους.
5. `keras.metrics`: Για την παρακολούθηση της απόδοσης κατά την εκπαίδευση.
6. `keras.callbacks`: Για τη ρύθμιση της εκπαίδευσης (π.χ., early stopping).

Δημιουργία και Εκπαίδευση Νευρωνικού Δικτύου

1. Διαδοχικό Μοντέλο (Sequential Model)

Το Sequential API είναι ο πιο απλός τρόπος δημιουργίας ενός μοντέλου, όπου τα επίπεδα προστίθενται το ένα μετά το άλλο.

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

# Δημιουργία ενός διαδοχικού μοντέλου
model = Sequential([
    Dense(32, activation='relu', input_shape=(10,)), # Πρώτο επίπεδο
    Dense(16, activation='relu'),                     # Δεύτερο επίπεδο
    Dense(1, activation='sigmoid')                    # Έξοδος
])

# Σύνοψη του μοντέλου
model.summary()

# Σύνθεση του μοντέλου
model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])

# Τεχνητά δεδομένα
```

```
import numpy as np
X = np.random.rand(100, 10) # 100 δείγματα, 10 χαρακτηριστικά
y = np.random.randint(0, 2, 100) # 0 ή 1 (δυναδική ταξινόμηση)

# Εκπαίδευση του μοντέλου
history = model.fit(X, y, epochs=10, batch_size=8, validation_split=0.2)
```

2. Λειτουργικό API (Functional API)

Το Functional API προσφέρει περισσότερη ευελιξία για τη δημιουργία σύνθετων αρχιτεκτονικών, όπως διακλαδώσεις ή συνένωση επιπέδων.

```
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input, Dense, concatenate

# Ορισμός εισόδου
input1 = Input(shape=(10,))
input2 = Input(shape=(5,))

# Κλάδος 1
x1 = Dense(32, activation='relu')(input1)

# Κλάδος 2
x2 = Dense(16, activation='relu')(input2)

# Συνένωση των δύο κλάδων
merged = concatenate([x1, x2])

# Τελικό επίπεδο
output = Dense(1, activation='sigmoid')(merged)

# Δημιουργία του μοντέλου
model = Model(inputs=[input1, input2], outputs=output)

# Σύνθεση και εκπαίδευση του μοντέλου
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

3. Σύνοψη Μοντέλου: model.summary()

Παρέχει μια σύνοψη της αρχιτεκτονικής του μοντέλου, συμπεριλαμβανομένων του αριθμού των παραμέτρων.

```
model.summary()
```

Προχωρημένες Δυνατότητες του Keras

4. Callbacks

Χρησιμοποιούνται για τον έλεγχο της εκπαίδευσης, π.χ., EarlyStopping, ModelCheckpoint.

```
from tensorflow.keras.callbacks import EarlyStopping

# Early stopping
early_stop = EarlyStopping(monitor='val_loss', patience=3)

# Εκπαίδευση με callback
model.fit(X, y, epochs=20, batch_size=8, validation_split=0.2, callbacks=[early_stop])
```

5. Αποθήκευση και Φόρτωση Μοντέλου

Αποθήκευση

```
# Αποθήκευση σε αρχείο  
model.save("my_model.h5")
```

Φόρτωση

```
from tensorflow.keras.models import load_model  
  
# Φόρτωση του μοντέλου  
model = load_model("my_model.h5")
```

6. Προεπεξεργασία Δεδομένων

Το Keras παρέχει εργαλεία για την προεπεξεργασία δεδομένων, όπως κανονικοποίηση ή tokenization για NLP.

```
from tensorflow.keras.preprocessing.text import Tokenizer  
  
# Δημιουργία tokenizer  
texts = ["Hello world", "Deep learning with Keras"]  
tokenizer = Tokenizer(num_words=10)  
tokenizer.fit_on_texts(texts)  
  
# Μετατροπή σε ακολουθίες  
sequences = tokenizer.texts_to_sequences(texts)  
print(sequences)
```

7. Αξιολόγηση Μοντέλου: `model.evaluate()`

Μετά την εκπαίδευση, αξιολογούμε το μοντέλο με ένα σύνολο δοκιμών.

```
loss, accuracy = model.evaluate(X_test, y_test)  
print(f"Loss: {loss}, Accuracy: {accuracy}")
```

8. Πρόβλεψη: `model.predict()`

Χρησιμοποιείται για να κάνουμε προβλέψεις με το εκπαιδευμένο μοντέλο.

```
# Πρόβλεψη σε νέα δεδομένα  
new_data = np.random.rand(5, 10) # 5 νέα δείγματα  
predictions = model.predict(new_data)  
print(predictions)
```

Υποστηριζόμενα Layers στο Keras

1. **Dense**: Πλήρως συνδεδεμένο επίπεδο.
2. **Dropout**: Αποτροπή υπερπροσαρμογής (overfitting).
3. **Conv2D**: Συνελικτικά επίπεδα για εικόνες.
4. **LSTM**: Long Short-Term Memory για ακολουθίες.
5. **Embedding**: Ενσωμάτωση για NLP.

Σύνοψη

Το **Keras** είναι ένα ευέλικτο εργαλείο για την ανάπτυξη και εκπαίδευση μοντέλων βαθιάς μάθησης. Προσφέρει μια απλή και κατανοητή διεπαφή για αρχάριους, ενώ είναι αρκετά ισχυρό για την υλοποίηση πολύπλοκων αρχιτεκτονικών. Χρησιμοποιώντας το Keras, μπορείτε να δημιουργήσετε και να πειραματιστείτε με διαφορετικούς αλγορίθμους μηχανικής μάθησης, εστιάζοντας στην ανάπτυξη και τη βελτίωση των μοντέλων σας.