

Bab 10: Introduction to Artificial Neural Networks with Keras

Tujuan Bab

Memahami dasar-dasar jaringan saraf tiruan (*artificial neural networks / ANN*), serta membangun dan melatih model ANN sederhana menggunakan Keras (API tingkat tinggi untuk TensorFlow).

Konsep Utama

1. Artificial Neuron

Model matematis terinspirasi dari neuron biologis. Sebuah neuron:

- Menerima input x_1, x_2, \dots, x_n
- Mengalikan tiap input dengan bobot w_i
- Menjumlahkan hasilnya, ditambah bias
- Melewatkan nilai akhir ke fungsi aktivasi

Contoh aktivasi:

- ReLU: umum digunakan karena mengurangi masalah vanishing gradient
 $\text{ReLU}(x) = \max(0, x)$
- Sigmoid: cocok untuk output probabilistik (biner)
- Tanh: mengoutputkan antara -1 dan 1

2. Architecture: Layers

Sebuah jaringan saraf terdiri dari:

- Input layer
- Hidden layers (bisa banyak)
- Output layer

Setiap layer berisi banyak neuron. Jika semua neuron dari satu layer terhubung ke semua neuron layer berikutnya → disebut fully connected / dense layer.

3. Building a Model with Keras

Contoh membuat model untuk klasifikasi:

```
from tensorflow import keras

model = keras.models.Sequential([
    keras.layers.Dense(30, activation="relu", input_shape=[8]),
    keras.layers.Dense(1, activation="sigmoid")
])
```

Model di atas terdiri dari:

- 1 hidden layer dengan 30 unit dan ReLU
- 1 output layer dengan 1 unit dan sigmoid (cocok untuk klasifikasi biner)

4. Kompilasi dan Pelatihan

Sebelum dilatih, model perlu dikompilasi:

```
model.compile(loss="binary_crossentropy", optimizer="sgd", metrics=["accuracy"])

Kemudian dilatih:

history = model.fit(X_train, y_train, epochs=20, validation_data=(X_valid, y_valid))
```

5. Evaluasi dan Prediksi

Model dapat dievaluasi dan digunakan untuk prediksi seperti berikut:

```
model.evaluate(X_test, y_test)

model.predict(X_new)
```

6. Saving and Loading Models

Model dapat disimpan dan dimuat kembali:

```
model.save("my_model.h5")  
  
model = keras.models.load_model("my_model.h5")
```

7. Callbacks

Keras mendukung callback seperti:

- ModelCheckpoint: menyimpan model terbaik selama pelatihan
- EarlyStopping: menghentikan pelatihan jika validasi tidak membaik

```
early_stopping_cb = keras.callbacks.EarlyStopping(patience=5,  
restore_best_weights=True)  
history = model.fit(..., callbacks=[early_stopping_cb])
```

Proyek / Notebook Praktik

Dataset

Menggunakan California Housing dataset untuk regresi.

Langkah Praktik:

1. Normalisasi data input
2. Membangun model Keras menggunakan Sequential
3. Melatih model dengan optimasi MSE dan menggunakan SGD
4. Menyimpan dan memuat model
5. Menambahkan callback EarlyStopping untuk menghindari overfitting

Inti Pelajaran

Konsep	Penjelasan
--------	------------

Neuron	Fungsi matematis yang memetakan input menjadi output nonlinier
Layer	Kumpulan neuron, dapat di-stack untuk membuat jaringan dalam
Keras Sequential	API sederhana untuk membuat model feedforward
Optimizer	Digunakan untuk menyesuaikan bobot (misalnya: SGD, Adam)
Loss Function	Mengukur seberapa jauh prediksi dari target (MSE, binary crossentropy)
Callback	Mekanisme otomatis selama pelatihan untuk efisiensi dan kestabilan

Keunggulan Keras

- Mudah digunakan
- Terintegrasi penuh dengan TensorFlow
- Dukungan penuh untuk GPU, checkpoint, dan callbacks
- Cocok untuk prototyping cepat maupun produksi