وزارة التعليم العالي والبحث العلمي
جامعة أم البواقي
كلية العلوم الاقتصادية والعلوم التجارية وعلوم التسيير
قسم: العلوم الاقتصادية **مقياس:** تطبيقات التعلم العميق

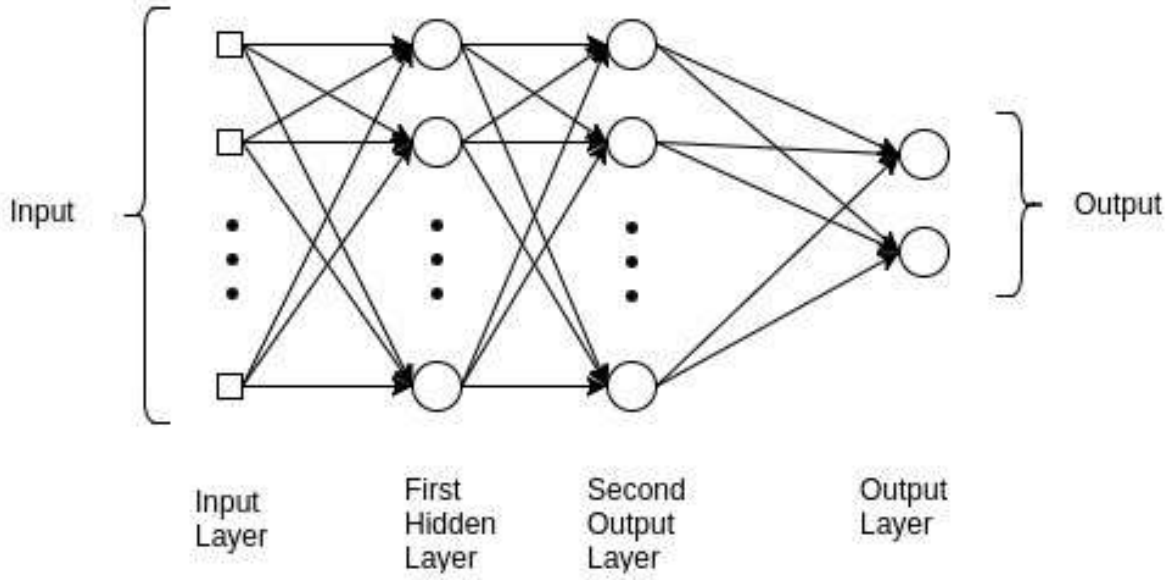
الدرس 03: التعرف على مكتبة Keras وTensorFlow

الأهداف:

1. فهم العلاقة بين TensorFlow وKeras.
2. معرفة أهم مكوّنات بناء النماذج العصبية في Keras.
3. بناء النموذج الأول باستخدام نموذج Sequential.
4. معرفة كل التعليمات الأساسية لإنشاء الطبقات، اختيار دوال التنشيط، إعداد التدريب وتقييمه.
5. تطبيق نموذج بسيط مرتبط بالاقتصاد والمحاسبة.

1. بنية شبكة عصبية متعددة الطبقات MLP

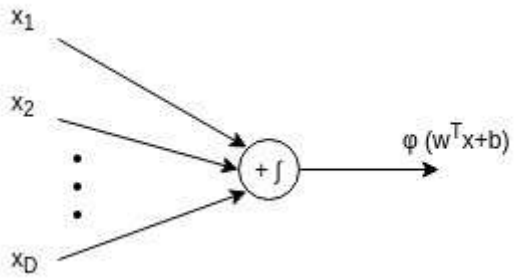
طبقة المدخل الطبقة المخفية 1 الطبقة المخفية 2 طبقة المخرج



الشبكة العصبية ذو المستقبلات متعددة الطبقات (Multilayer Perceptron Neural Network **MLP**)

تتكون كل طبقة من مجموعة من العقد أو العصبونات المتصلة بالعقد الموجودة في الطبقات المجاورة .يتم ترجيح الروابط بين العقد في شبكة MLP ، ويتم تعليم هذه الأوزان أثناء عملية التدريب .الهدف من تدريب شبكة MLP هو ضبط الأوزان بطريقة تمكن النموذج من التنبؤ بدقة ببيانات الإخراج لمدخل معين .إذا كنت على دراية بخوارزمية المستقلات Perceptron Algorithm ، فإننا في المستقبلات فقط نضرب الأوزان Weights في المدخلات Inputs و تضيف الانحياز Bias ، لكننا نقوم بذلك فقط في طبقة واحدة. حيث تقوم الشبكة بتحديث الاوزان عندما تجد خطأ في التصنيف. معادلة تحديث الوزن هي كالتالي

الوزن weight = الوزن السابق weight + معدل التعلم learning_rate*(القيمة المتوقعه expected – القيمة التي تم ايجادها predicted) * قيمة المدخل x



Perceptron = Neurons = عصبون

2. ما هي TensorFlow وKeras؟

- **Keras** = واجهة عالية المستوى

وهي طبقة بسيطة ومرنة تساعد في بناء الشبكات العصبية بسهولة.

- **TensorFlow** = المحرك الخلفي

يقوم بتنفيذ:

- العمليات الرياضية(Matrix Multiplications)
- التدريبBackpropagation
- تشغيل النماذج على GPU و TPU (Tensor Processing Unit)

لذلك:

Keras = طريقة الكتابة | TensorFlow = طريقة التنفيذ

3. مكونات النموذج العصبي في Keras

I. أنواع النماذج Models

a. النموذج المتسلسل (Sequential Model)

الأكثر شيوعًا وسهولة:

يُستخدم عندما تكون الطبقات متتالية بشكل خطّي.

```
from tensorflow.keras.models import Sequential
model = Sequential()
```

b. النموذج الوظيفي Functional API

يُستخدم في شبكات أكثر تعقيدًا (ليست جزءًا من هذا الأسبوع عادة).

مثال:

II. الطبقات (Layers)

a. الطبقة Dense (الأكثر استخدامًا)

وتسمى "الطبقة الكاملة الاتصال" Fully Connected Layer:

- هي المكونات الأساسية للشبكة العصبية.
- كل طبقة تحتوي على مجموعة من "العُقد العصبية."(Neurons)

b. Dropout Layer — لمنع الإفراط في التعلّم

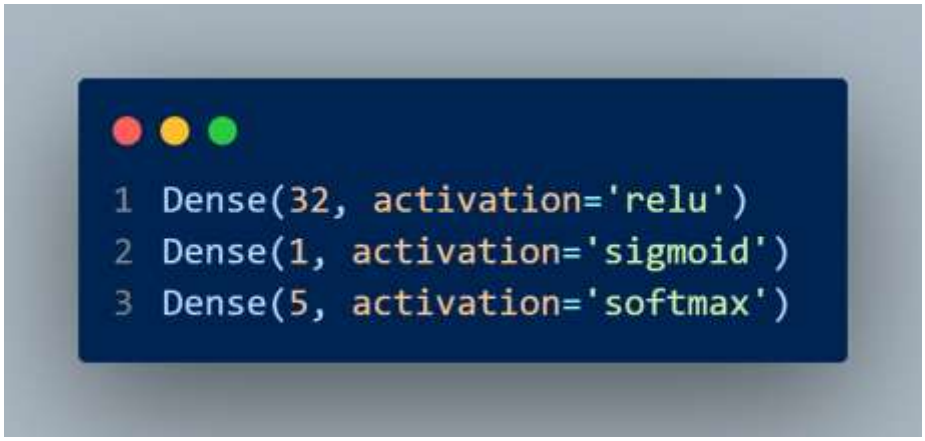
تُستخدم لتقليل Overfitting.

III. دوال التنشيط (Activation Functions)

الأكثر استخدامًا:

الدالة	الاستخدام
relu	غالبية الشبكات العميقة
sigmoid	تصنيف ثنائي (1/0)
softmax	تصنيف متعدد
tanh	بديل للـ sigmoid
linear	انحدار (Regression)

أمثلة:



```
Dense(32, activation='relu')
Dense(1, activation='sigmoid')
Dense(5, activation='softmax')
```

IV. دوال الخطأ(Loss Functions)

نوع المشكلة	الدالة
تصنيف ثنائي	binary_crossentropy
تصنيف متعدد	categorical_crossentropy
انحدار	mean_squared_error (MSE)

مثال:

```
model.compile(loss='binary_crossentropy')
```

4. أهم أوامر التدريب

تدريب النموذج:

التنبؤ:

تقييم النموذج:

خلاصة:

مراحل بناء نموذج فيKeras

1. **تحديد النموذج :**Sequential أو Functional.
2. **إضافة الطبقات.(Dense Layers)**
3. **تجميع النموذج :(Compile)** تحديد الخسارة والمقياس وخوارزمية التحسين.
4. **تدريب النموذج :(Fit)** تمرير البيانات ليتعلم منها.
5. **التنبؤ :(Predict)** اختبار النموذج على بيانات جديدة.

```python
import numpy as np
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

# بيانات
X = np.array([
    [5, 50],
    [10, 120],
    [15, 180],
    [20, 240],
    [25, 300]
], dtype=float)

y = np.array([[100000], [180000], [250000], [320000], [400000]])

model = Sequential([
    Dense(16, activation='relu', input_shape=(2,)),
    Dense(8, activation='relu'),
    Dense(1)  # انحدار → لا نحتاج دالة تنشيط
])

model.compile(optimizer='adam', loss='mean_squared_error')

model.fit(X, y, epochs=800, verbose=0)

# التنبؤ
print(model.predict([[30, 350]]))
```