
1. **_____**

_____ **_____** _____-_____ **_____** _____
_____ **_____** _____, **_____** _____ **_____** _____
_____ **_____** _____, **_____** _____ **_____** _____.
_____ **_____** _____, **_____** _____ **_____** _____
_____ **_____** _____, **_____** _____ **_____** _____
_____.

_____ **_____**: **_____** _____ **_____** **_____** _____
_____ **_____** _____ **_____** _____ **_____** _____ (CNN) **_____** **_____**
_____ **_____**.

2. **_____**

_____ — **_____** **_____** **_____**, **_____**
_____ **_____** **_____** **_____** **_____** CNN **_____**
_____ (convolutional layers), **_____** **_____** (pooling layers), **_____**
_____'_____ **_____** (fully connected layers). **_____** **_____**
_____ **_____** **_____** **_____** **_____**, **_____**
_____.

_____ **_____** **_____** **_____** **_____** (forward
propagation), **_____** **_____** **_____** **_____** **_____**
_____ (backpropagation). **_____** **_____** **_____** Python: TensorFlow,
Keras, PyTorch.

3. **_____**

_____ **_____** **_____** **_____** **_____** CNN-**_____** **_____**
_____ **_____** **_____** **_____**. **_____** **_____** **_____** **_____**
_____ **_____** **_____**, **_____** **_____**, **_____** **_____** **_____**
_____ **_____** **_____**, **_____** **_____** **_____** **_____**
_____.

accuracy (accuracy) ■ loss (loss) ■
.

4. **Preprocess Data**

首先，我们将使用 `ImageDataGenerator` 对数据进行预处理。这将自动完成归一化、裁剪和旋转等操作。我们还将使用 `flow_from_directory` 方法从目录中加载数据。对于训练集，我们将使用 `train_datagen.flow_from_directory`，对于验证集，我们将使用 `validation_datagen.flow_from_directory`。Keras 的 `ImageDataGenerator`。

5. **Train Model**

现在，我们可以使用 `fit_generator` 方法训练模型。我们将使用训练集生成器和验证集生成器，并设置一些参数，如批大小、训练步数和验证步数。我们还将监控训练进度，并在每一步之后保存模型。这样，我们可以在训练过程中随时恢复模型，而不会丢失任何进度。

6. [REDACTED]
1. [REDACTED]
[REDACTED] [REDACTED] [REDACTED] [REDACTED] ([REDACTED], CIFAR-10
[REDACTED] MNIST). 2. [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED]
[REDACTED] [REDACTED]. 3. [REDACTED] [REDACTED] [REDACTED]
[REDACTED] [REDACTED] [REDACTED] [REDACTED].