

Report

בעבור הארכיטקטורה השתמשתי ברשת נוירונים CNN עם fully connected. לאחר סקירה באינטרנט ודוגמאות שראינו בהרצאה, הבנתי שנצטרך כמה שכבות של CNN ולכן הרצתי כמה שכבות עד שהגעתי לתוצאה הרצויה עם סיכויי הצלחה גבוהים.

היפר פרמטרים:

Epoch:

Epoch: 1, Accuracy: (85%)

Epoch: 2, Accuracy: (89%)

Epoch: 3, Accuracy: (90%)

Epoch: 4, Accuracy: (92%)

Epoch: 5, Accuracy: (88%)

Learning_Rate:

Learning_Rate: 0.01, Accuracy: (80%)

Learning_Rate: 0.005, Accuracy: (85%)

Learning_Rate: 0.001, Accuracy: (90%)

Learning_Rate: 0.0005, Accuracy: (83%)

Batch_Size:

Batch_Size: 50, Accuracy: (85%)

Batch_Size: 100, Accuracy: (91%)

Batch_Size: 150, Accuracy: (87%)

Batch_Size: 300, Accuracy: (85%)

לכן לאחר בדיקות הגעתי להיפר פרמטרים הבאים:

EPOCHS = 4

LEARNING_RATE = 0.001 – ניסינו גם 0.01, 0.05, 0.005

BATCH_SIZE = 100 – ניסינו גם 50, 150, 300

נשים לב שהרצנו כמה איפוקים וכאשר ראינו שאחוז ה validation ירד לקחנו את מ'ס האיפוקים המקסימאלי.

ארכיטקטורה:

עשינו 5 שכבות של קונבולוציה CNN ולאחר מכן Fully Connected

שכבה 1:

- קונבולוציה מערוץ בניסה 1 ל- 32 , $stride = 1$, $kernal = 5$, ועם שתי שכבות של padding .
- Leaky Relu עם קבוע 0.2 .
- Max Pooling עם $stride = 2$, $kernal = 2$.

שכבה 2:

- קונבולוציה עם 32 ערצי בניסה ל- 64 , $stride = 1$, $kernal = 5$, ועם שתי שכבות של padding .
- Leaky Relu עם קבוע 0.2 .
- Max Pooling עם $stride = 2$, $kernal = 2$.

שכבה 3:

- קונבולוציה עם 64 ערצי בניסה ל- 128 , $stride = 1$, $kernal = 5$, ועם שתי שכבות של padding .
- Leaky Relu עם קבוע 0.2 .
- Max Pooling עם $stride = 2$, $kernal = 2$.

שכבה 4:

- קונבולוציה עם 128 ערצי בניסה ל- 64 , $stride = 1$, $kernal = 5$, ועם שתי שכבות של padding .
- Leaky Relu עם קבוע 0.2 .
- Max Pooling עם $stride = 2$, $kernal = 2$.

שכבה 5:

- קונבולוציה עם 64 ערצי בניסה ל- 32 , $stride = 1$, $kernal = 5$, ועם שתי שכבות של padding .
- Leaky Relu עם קבוע 0.2 .
- Max Pooling עם $stride = 2$, $kernal = 2$.

Fully Connected:

- שיטחתי את המטריצה לוקטור בגודל 480 . ואז ביצעתי 2 שכבות נסתרות.
- שכבה ראשונה מ 480 נוירונים ל- 520 .
- Dropout עם $p=0.4$.
- Leaky Relu עם קבוע 0.2 .
- שכבה שנייה מ- 512 נוירונים ל- 256 .
- Dropout עם $p=0.4$.
- Leaky Relu עם קבוע 0.2 .

ולבסוף כדי להגיע לשכבת סיום מ- 215 ל- 30.

בסוף מבצעים Log Softmax כדי להחזיר את ההסתברות לכל מחלקה.
לבסוף השתמשתי באופטימיזר של ADAM כמו שלמדנו בהרצאה שהוא
אמור לתת את התוצאות הטובות ביותר.

[המודל בקוד:](#)

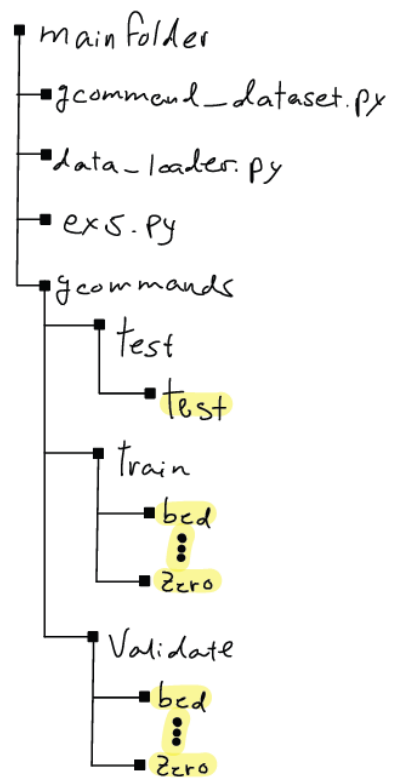
```
def __init__(self):
    super(Model, self).__init__()
    # layer 1
    self.layer1 = Sequential(
        Conv2d(1, 32, kernel_size=5, stride=1, padding=2),
        LeakyReLU(0.2, inplace=True),
        MaxPool2d(kernel_size=2, stride=2))
    # layer 2
    self.layer2 = Sequential(
        Conv2d(32, 64, kernel_size=5, stride=1, padding=2),
        LeakyReLU(0.2, inplace=True),
        MaxPool2d(kernel_size=2, stride=2))
    # layer 3
    self.layer3 = Sequential(
        Conv2d(64, 128, kernel_size=5, stride=1, padding=2),
        LeakyReLU(0.2, inplace=True),
        MaxPool2d(kernel_size=2, stride=2))
    # layer 4
    self.layer4 = Sequential(
        Conv2d(128, 64, kernel_size=5, stride=1, padding=2),
        LeakyReLU(0.2, inplace=True),
        MaxPool2d(kernel_size=2, stride=2))
    # layer 5
    self.layer5 = Sequential(
        Conv2d(64, 32, kernel_size=5, stride=1, padding=2),
        LeakyReLU(0.2, inplace=True),
        MaxPool2d(kernel_size=2, stride=2))
    # fully_connected
    self.fully_connected = Sequential(
        Linear(480, 512),
        Dropout(0.4),
        LeakyReLU(0.2, inplace=True),
        Linear(512, 256),
        Dropout(0.4),
        LeakyReLU(0.2, inplace=True),
        Linear(256, 30))
```

הרצה:

הדרך הרצה היא כמו הרגילה : python ex5.py כאשר נמצאים בתיקה שבה נמצא
ex5.py.

היררכיה בתיקיות:

אצלנו ההיררכיה היא כזו:



בעוד שהתקיות הצהובות מכילות קבצי wav.