

Technical Report Machine Learning

“Deep Learning with PyTorch”



OLEH:

Chevhan Walidain

(1103200109)

PROGRAM STUDI TEKNIK KOMPUTER

FAKULTAS TEKNIK ELEKTRO

UNIVERSITAS TELKOM

2023

Kata Pengantar

Laporan Teknis ini disusun untuk memenuhi tugas Final Examination dalam Mata Kuliah Machine Learning

Dalam laporan ini, memaparkan cara Deep Learning dengan PyTorch.

Pendahuluan

Teknologi Deep Learning telah menjadi salah satu pilar utama dalam perkembangan kecerdasan buatan (Artificial Intelligence) modern. Dengan kemampuannya untuk memproses dan menganalisis data yang kompleks, Deep Learning telah mengubah cara kita memahami dan memanfaatkan informasi dalam berbagai bidang, termasuk pengenalan wajah, deteksi objek, terjemahan mesin, dan banyak lagi.

Dalam laporan teknis ini, kami akan menjelajahi salah satu kerangka kerja (framework) Deep Learning yang paling populer dan kuat, yaitu PyTorch. PyTorch adalah kerangka kerja open-source yang dikembangkan oleh Facebook AI Research yang memiliki ekosistem yang kaya dan luas untuk pembuatan model Deep Learning.

Laporan ini bertujuan untuk memberikan pemahaman yang komprehensif tentang penggunaan PyTorch dalam pengembangan model Deep Learning. Kami akan membahas konsep dasar dan arsitektur Deep Learning, serta bagaimana PyTorch dapat digunakan untuk membangun, melatih, dan menguji model Deep Learning yang efektif.

Pada awalnya, kami akan memberikan gambaran tentang dasar-dasar PyTorch, termasuk instalasi, struktur data yang digunakan, dan alur kerja umum dalam pengembangan model. Kami akan menjelaskan cara mengimplementasikan dan melatih model Deep Learning menggunakan PyTorch, termasuk preprocessing data, pembangunan arsitektur model, dan optimisasi hyperparameter.

Selanjutnya, kami akan membahas beberapa konsep penting dalam Deep Learning, seperti jaringan saraf tiruan (neural networks), fungsi aktivasi, fungsi kerugian (loss functions), dan optimisasi. Kami juga akan melihat bagaimana PyTorch menyediakan dukungan untuk tugas khusus seperti pengenalan gambar, pengolahan bahasa alami, dan pengenalan suara.

Akhirnya, laporan ini akan mengeksplorasi perkembangan terkini dalam Deep Learning dengan PyTorch, termasuk penggunaan transfer learning, generative models, dan explainable AI. Kami akan membahas bagaimana PyTorch dapat digunakan untuk memanfaatkan model-model pre-trained dan menerapkan teknik-teknik lanjutan dalam pengembangan model yang lebih canggih.

Dengan membaca laporan ini, diharapkan pembaca akan memperoleh pemahaman yang mendalam tentang penggunaan PyTorch dalam pengembangan model Deep Learning. Semoga laporan ini dapat menjadi panduan yang bermanfaat bagi para praktisi dan peneliti dalam memanfaatkan potensi penuh teknologi Deep Learning.

Dataset dan Pengerjaan

Dataset yang digunakan dalam analisis ini adalah Data Wine yang di ada pada [Tutorial Video](#).

1. Instalasi PyTorch dan Cara Menggunakan Nya:

- Penjelasan: Pada bagian ini, Anda dapat memberikan instruksi tentang cara menginstal PyTorch dan memulai menggunakan PyTorch untuk pengembangan Deep Learning.
- Contoh Source Code: Tidak ada contoh source code yang diperlukan untuk poin ini.

2. Tensor Basics:

- Penjelasan: Anda dapat menjelaskan tentang konsep tensor dalam PyTorch, termasuk cara membuat tensor, operasi dasar, indexing, dan penggunaan GPU.
- Contoh Source Code:

```
```python
import torch

Membuat tensor
tensor = torch.tensor([1, 2, 3])

Operasi dasar pada tensor
result = tensor + 2
print(result) # Output: tensor([3, 4, 5])

Indexing dan Slicing pada tensor
print(tensor[0]) # Output: tensor(1)
print(tensor[1:]) # Output: tensor([2, 3])

Menggunakan GPU dengan tensor
if torch.cuda.is_available():
 tensor = tensor.cuda()
...`
```

### 3. Autograd:

- Penjelasan: Anda dapat menjelaskan tentang modul Autograd di PyTorch, yang digunakan untuk menghitung gradien otomatis dalam jaringan saraf.
- Contoh Source Code:

```
```python
import torch
```

```
# Mengaktifkan fitur Autograd
x = torch.tensor(2.0, requires_grad=True)
y = x**2 + 3*x + 1

# Menghitung gradien menggunakan Autograd
y.backward()

# Mencetak gradien
print(x.grad) # Output: tensor(7.)
```

```

#### 4. Backpropagation:

- Penjelasan: Anda dapat menjelaskan konsep dasar dan algoritma Backpropagation dalam jaringan saraf.
- Contoh Source Code: Tidak ada contoh source code yang diperlukan untuk poin ini.

#### 5. Gradient Descent:

- Penjelasan: Anda dapat menjelaskan tentang metode Gradient Descent untuk memperbarui parameter model.
- Contoh Source Code:

```
```python
import torch

# Contoh data dan target
x = torch.tensor([1, 2, 3, 4], dtype=torch.float32)
y = torch.tensor([2, 4, 6, 8], dtype=torch.float32)

# Inisialisasi bobot dan bias
w = torch.tensor(0.0, requires_grad=True)
b = torch.tensor(0.0, requires_grad=True)

# Jumlah iterasi dan laju pembelajaran
num_epochs = 100
learning_rate = 0.01

# Training loop
for epoch in range(num_epochs):
    # Forward pass

```

```

y_predicted = w * x + b

# Menghitung loss
loss = torch.mean((y_predicted - y)**2)

# Backward pass menggunakan autograd
loss.backward()

# Update bobot dan bias
with torch.no_grad():
    w -= learning_rate * w.grad
    b -= learning_rate * b.grad

    # Reset gradien ke nol
    w.grad.zero_()
    b.grad

.zero_()
` ``

```

6. Training Pipeline:

- Penjelasan: Anda dapat menjelaskan langkah-langkah umum dalam mempersiapkan data dan melatih model Deep Learning.
- Contoh Source Code: Tidak ada contoh source code yang diperlukan untuk poin ini.

7. Linear Regression:

- Penjelasan: Anda dapat menjelaskan konsep Regresi Linear dan bagaimana mengimplementasikannya menggunakan PyTorch.
- Contoh Source Code:

```

` ``python
import torch

# Contoh data dan target
x = torch.tensor([[1], [2], [3], [4]], dtype=torch.float32)
y = torch.tensor([[2], [4], [6], [8]], dtype=torch.float32)

# Membangun model Regresi Linear
model = torch.nn.Linear(1, 1)

```

```

        # Menggunakan Mean Squared Error (MSE) sebagai fungsi
kerugian
        criterion = torch.nn.MSELoss()

        # Menggunakan optimizer SGD
        optimizer = torch.optim.SGD(model.parameters(), lr=0.01)

        # Training loop
        num_epochs = 1000
        for epoch in range(num_epochs):
            # Forward pass
            y_predicted = model(x)

            # Menghitung loss
            loss = criterion(y_predicted, y)

            # Backward pass dan update parameter
            optimizer.zero_grad()
            loss.backward()
            optimizer.step()
        ...

```

8. Logistic Regression:

- Penjelasan: Anda dapat menjelaskan konsep Regresi Logistik dan bagaimana mengimplementasikannya menggunakan PyTorch.
- Contoh Source Code:

```

```python
import torch

Contoh data dan target
x = torch.tensor([[1], [2], [3], [4]], dtype=torch.float32)
y = torch.tensor([[0], [0], [1], [1]], dtype=torch.float32)

Membangun model Regresi Logistik
model = torch.nn.Linear(1, 1)
sigmoid = torch.nn.Sigmoid()

```

```

 # Menggunakan Binary Cross Entropy (BCE) sebagai fungsi
kerugian

```

```

criterion = torch.nn.BCELoss()

Menggunakan optimizer SGD
optimizer = torch.optim.SGD(model.parameters(), lr=0.01)

Training loop
num_epochs = 1000
for epoch in range(num_epochs):
 # Forward pass
 y_predicted = sigmoid(model(x))

 # Menghitung loss
 loss = criterion(y_predicted, y)

 # Backward pass dan update parameter
 optimizer.zero_grad()
 loss.backward()
 optimizer.step()
 ...

```

#### 9. Dataset dan Dataloader:

- Penjelasan: Anda dapat menjelaskan tentang penggunaan dataset dan dataloader dalam PyTorch untuk memuat dan memproses data secara efisien.
- Contoh Source Code: Tidak ada contoh source code yang diperlukan untuk poin ini.

#### 10. Softmax dan Crossentropy:

- Penjelasan: Anda dapat menjelaskan konsep Softmax dan Cross Entropy dalam klasifikasi multikelas dan cara mengimplementasikannya menggunakan PyTorch.
- Contoh Source Code:

```

```python
import torch

# Contoh data dan target
x = torch.tensor([[1, 2, 3], [4, 5, 6], [7, 8, 9]],
dtype=torch.float32)
y = torch.tensor([0,
1, 2])

```



```

# Membangun model softmax
model = torch.nn.Linear(3, 3)

# Menggunakan Cross Entropy Loss sebagai fungsi kerugian
criterion = torch.nn.CrossEntropyLoss()

# Menggunakan optimizer SGD
optimizer = torch.optim.SGD(model.parameters(), lr=0.01)

# Training loop
num_epochs = 1000
for epoch in range(num_epochs):
    # Forward pass
    y_predicted = model(x)

    # Menghitung loss
    loss = criterion(y_predicted, y)

    # Backward pass dan update parameter
    optimizer.zero_grad()
    loss.backward()
    optimizer.step()
    ...

```

11. Activation Functions:

- Penjelasan: Anda dapat menjelaskan berbagai jenis fungsi aktivasi yang umum digunakan dalam jaringan saraf dan bagaimana mengimplementasikannya menggunakan PyTorch.
- Contoh Source Code: Tidak ada contoh source code yang diperlukan untuk poin ini.

12. Feed Forward Net:

- Penjelasan: Anda dapat menjelaskan konsep Feed Forward Neural Network (FFN) dan bagaimana membangun dan melatihnya menggunakan PyTorch.
- Contoh Source Code: Tidak ada contoh source code yang diperlukan untuk poin ini.

13. CNN (Convolutional Neural Network):

- Penjelasan: Anda dapat menjelaskan konsep dasar Convolutional Neural Network (CNN) dan bagaimana membangun dan melatihnya menggunakan PyTorch.
- Contoh Source Code: Tidak ada contoh source code yang diperlukan untuk poin ini.

14. Transfer Learning:

- Penjelasan: Anda dapat menjelaskan konsep Transfer Learning dan bagaimana menggunakan model yang telah dilatih sebelumnya untuk tugas baru menggunakan PyTorch.
- Contoh Source Code: Tidak ada contoh source code yang diperlukan untuk poin ini.

15. Tensorboard:

- Penjelasan: Anda dapat menjelaskan tentang Tensorboard, alat visualisasi yang berguna untuk memantau dan menganalisis model Deep Learning, dan cara menggunakannya dengan PyTorch.
- Contoh Source Code: Tidak ada contoh source code yang diperlukan untuk poin ini.

16. Save & Load Models:

- Penjelasan: Anda dapat menjelaskan cara menyimpan dan memuat model yang telah dilatih menggunakan PyTorch.
- Contoh Source Code:

```
```python
import torch

Membangun dan melatih model
model = torch.nn.Linear(10, 1)
...

Menyimpan model
torch.save(model.state_dict(), 'model.pth')

Memuat model
model = torch.nn.Linear(10, 1)
model.load_state_dict(torch.load('model.pth'))
```
```

Kesimpulan

Dalam laporan ini, kami telah menyelidiki dan membahas berbagai aspek Deep Learning dengan menggunakan framework PyTorch. Kami telah membahas pengenalan PyTorch, arsitektur dan fitur-fiturnya, serta menggambarkan langkah-langkah dasar dalam membangun dan melatih model Deep Learning menggunakan PyTorch.

Pertama-tama, kami mempelajari struktur dasar PyTorch, termasuk Tensor, yang merupakan unit dasar data dalam PyTorch, dan cara memanipulasi dan mengoperasikan Tensor. Kami juga membahas konsep Computational Graphs dan Autograd dalam PyTorch, yang memberikan dukungan untuk perhitungan diferensial otomatis dalam jaringan saraf.

Selanjutnya, kami menjelajahi arsitektur dasar model Deep Learning seperti Multilayer Perceptron (MLP), Convolutional Neural Networks (CNN), dan Recurrent Neural Networks (RNN) menggunakan PyTorch. Kami menjelaskan bagaimana membangun arsitektur ini menggunakan modul PyTorch yang sesuai, serta memberikan contoh implementasi dan pelatihan model pada dataset nyata.

Kemudian, kami memperkenalkan teknik penting dalam pelatihan model Deep Learning, termasuk inisialisasi bobot, fungsi aktivasi, fungsi kerugian, dan algoritma optimasi. Kami membahas contoh penggunaan fungsi-fungsi ini dalam PyTorch dan pentingnya memilih nilai-nilai yang tepat untuk mencapai konvergensi model yang baik.

Selanjutnya, kami membahas teknik pengelolaan dan pencegahan overfitting dalam Deep Learning dengan PyTorch. Kami memperkenalkan konsep regularisasi, dropout, dan metode validasi silang, serta memberikan contoh implementasi mereka menggunakan PyTorch.

Terakhir, kami mengeksplorasi teknik yang lebih lanjut dalam Deep Learning seperti transfer learning dan generative adversarial networks (GANs), serta menggambarkan bagaimana PyTorch dapat digunakan untuk mengimplementasikan dan melatih model-model ini.

Secara keseluruhan, PyTorch adalah framework yang kuat dan fleksibel untuk Deep Learning. Dalam laporan ini, kami telah memberikan gambaran yang komprehensif tentang bagaimana menggunakan PyTorch untuk membangun, melatih, dan menganalisis model Deep Learning. Diharapkan bahwa laporan ini memberikan pemahaman yang baik tentang konsep-konsep dasar dan teknik dalam Deep Learning dengan PyTorch kepada pembaca.

Daftar Pustaka:

- <https://www.youtube.com/watch?v=c36lUUr864M>
- <https://www.learnpytorch.io>