Vision Transformer (ViT) for Fashion-MNIST Classification: Professional Tutorial and Evaluation

GitHub: https://github.com/waseem1415/ViT-FashionMNIST

1. Introduction

Machine learning, particularly deep learning, has revolutionized computer vision over the past decade. Convolutional Neural Networks (CNNs) have traditionally dominated visual recognition tasks, achieving remarkable performance on standard benchmarks such as MNIST, CIFAR-10, and ImageNet. However, a new paradigm has emerged: the **Vision Transformer (ViT)**, which adapts the **Transformer architecture** from natural language processing (NLP) to image classification. ViTs leverage self-attention mechanisms to model global relationships across image patches, offering a fundamentally different approach compared to local convolutions in CNNs.

In this tutorial, we implement a **Vision Transformer on the Fashion-MNIST dataset**, a widely used benchmark for evaluating image classification algorithms. Fashion-MNIST contains 70,000 grayscale images of fashion items, including T-shirts, trousers, dresses, and shoes, each of size 28x28 pixels. The dataset is split into **60,000 training images** and **10,000 test images**, with 10 balanced categories. Despite its simplicity, Fashion-MNIST presents non-trivial challenges due to subtle differences between classes (e.g., shirt vs. coat or sneaker vs. sandal).

The primary objective of this tutorial is to demonstrate the **design, implementation, and evaluation of a ViT model**, with a focus on how patch size, depth, and attention mechanisms influence classification performance. This tutorial also emphasizes **reproducibility, accessibility, and professional practices**, making it suitable for students and practitioners seeking to adopt ViT for their own tasks.

2. Background and Literature Review

2.1 Transformers in Machine Learning

Transformers were introduced by Vaswani et al. (2017) in the seminal paper *"Attention is All You Need"* for NLP tasks. Unlike recurrent neural networks (RNNs) or CNNs, Transformers rely entirely on **self-attention mechanisms**, allowing each element in a sequence to interact with all others. This design enables the model to capture **long-range dependencies** efficiently and to be trained in parallel.

Key components of a Transformer include:

1. **Multi-Head Self-Attention (MHSA):** Captures relationships between all tokens in the input sequence by computing multiple attention distributions simultaneously.
2. **Layer Normalization:** Stabilizes training by normalizing inputs across features.
3. **Feed-Forward Networks (FFN):** Non-linear transformations applied to each token individually.
4. **Residual Connections:** Help mitigate vanishing gradient problems and enable deeper architectures.

2.2 Vision Transformer (ViT)

Dosovitskiy et al. (2020) introduced the **Vision Transformer**, adapting the Transformer for image classification:

- **Patch Embedding:** The input image is split into fixed-size patches (e.g., 4x4 or 16x16 pixels). Each patch is flattened and projected into a fixed-dimensional embedding.
- **Class Token:** A learnable vector prepended to the patch embeddings, whose representation after the Transformer encoder is used for classification.
- **Positional Encoding:** Adds spatial information to the patches, as Transformers are permutation-invariant by design.
- **Transformer Encoder:** Consists of multiple stacked layers of MHSA and FFN blocks.
- **MLP Head:** Final classification layer mapping the class token to output classes.

Advantages of ViT over CNNs include **scalability to large datasets**, **ability to model long-range dependencies**, and **flexibility in architecture design**, while disadvantages include **high data requirements** and sensitivity to patch size and hyperparameters.

2.3 Fashion-MNIST Dataset

Fashion-MNIST was introduced by Xiao et al. (2017) as a **drop-in replacement for MNIST**. It contains:

- 28x28 grayscale images
- 10 categories: T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle boot
- Balanced class distribution: 7,000 images per class (6,000 train, 1,000 test)

The dataset is challenging due to **subtle inter-class differences** and serves as an excellent benchmark for modern image classification methods.

3. Methodology

3.1 Data Preprocessing

We applied the following transformations:

1. **Tensor Conversion:** Convert images from PIL format to PyTorch tensors.
2. **Normalization:** Normalize pixel values to mean = 0.5, std = 0.5 to facilitate faster and more stable training.

The training dataset (60,000 images) was split into:

- **Train:** 55,000 images
- **Validation:** 5,000 images

Data loaders were created with **batch size = 64**, shuffling for training data to ensure model robustness.

3.2 Model Architecture

The implemented Vision Transformer consists of the following components:

1. **Patch Embedding Layer:**
    - Patch size = 4x4
    - Embedding dimension = 64
    - Each 28x28 image produces 49 patches (7x7 grid)
2. **Class Token & Positional Encoding:**
    - A learnable class token prepended to patch embeddings
    - Positional encoding added to each patch and class token to retain spatial information
3. **Transformer Encoder:**
    - Depth = 4 layers
    - Multi-head self-attention with 8 heads
    - Feed-forward network with hidden dimension = 128
    - Residual connections and layer normalization applied to all blocks

4. **MLP Head:**
    - LayerNorm followed by linear classification to 10 classes

The model is modular, making it easy to **adjust depth, patch size, attention heads, and FFN dimensions**.

3.3 Training Procedure

- **Loss Function:** Cross-entropy loss for multi-class classification
- **Optimizer:** Adam with learning rate = 0.001
- **Epochs:** 20
- **Device:** CPU/GPU detection implemented for portability
- **Reproducibility:** Random seeds set for PyTorch, NumPy, and Python random library

Training includes **validation evaluation at each epoch**, tracking:

- Training loss
- Validation loss
- Validation accuracy

This approach ensures **early detection of overfitting** and facilitates hyperparameter tuning.

4. Results

4.1 Training and Validation Performance

Epoch	Train Loss	Train Acc	Val Loss	Val Acc
1	0.7824	0.7125	0.5466	0.8084
10	0.3081	0.8855	0.3315	0.8780
20	0.2494	0.9060	0.3093	0.8908

Observations:

- Rapid decrease in training loss in early epochs indicates effective learning.
- Validation accuracy stabilizes around 89%, suggesting good generalization.
- Minimal overfitting observed due to consistent validation performance.

4.2 Confusion Matrix

The confusion matrix shows **per-class performance**:

- Highest accuracy: Trouser (98%), Sandal (97%), Bag (97%)
- Lowest accuracy: Shirt (65%), Pullover (86%)

Insights:

- Confusion mostly occurs among visually similar classes (e.g., Shirt vs. T-shirt/top, Coat vs. Pullover).
- This aligns with expectations due to **subtle visual differences** in Fashion-MNIST.

4.3 Classification Report

Class	Precision	Recall	F1-Score
T-shirt/top	0.84	0.83	0.83
Trouser	0.99	0.98	0.98
Pullover	0.79	0.86	0.82
Dress	0.87	0.91	0.89
Coat	0.81	0.79	0.80
Sandal	0.97	0.97	0.97
Shirt	0.73	0.65	0.69
Sneaker	0.93	0.96	0.95
Bag	0.97	0.98	0.97
Ankle boot	0.97	0.94	0.96

- **Macro average F1-score:** 0.89
- **Weighted average F1-score:** 0.89

The metrics confirm that the model is **highly effective across most classes**, with minor weaknesses in visually ambiguous classes.

4.4 Loss and Accuracy Curves

- Training and validation loss **decrease smoothly**, indicating stable learning.
- Validation accuracy gradually improves, reflecting **effective generalization**.
- Curves show **no significant overfitting**, suggesting regularization via dropout and proper model depth is effective.

5. Discussion

5.1 Model Strengths

- **Global attention:** ViT captures long-range dependencies, improving classification for items with distributed visual features.
- **Modular architecture:** Enables easy experimentation with depth, patch size, and attention heads.
- **High accuracy:** Achieves ~89% test accuracy, competitive for a moderate-sized dataset like Fashion-MNIST.

5.2 Limitations

- **Data requirements:** ViTs typically benefit from larger datasets. Fashion-MNIST is relatively small, limiting potential gains.
- **Computational cost:** Training Transformers is more resource-intensive than CNNs.
- **Confusion in similar classes:** Shirt vs. T-shirt/top or Coat vs. Pullover remain challenging.

5.3 Future Improvements

- **Data augmentation:** Can enhance generalization and reduce class confusion.
- **Hyperparameter tuning:** Experimenting with patch size, depth, attention heads, and learning rates.
- **Hybrid models:** Combining CNN feature extraction with ViT may improve performance on small datasets.
- **Pretraining on larger datasets:** Leveraging pretrained ViTs from ImageNet could improve accuracy.

6. Accessibility and Reproducibility

- **Random seeds:** Ensure deterministic results.
- **Color-blind friendly palettes:** Confusion matrices and plots designed for accessibility.
- **Modular code:** Easy to adapt and reuse for other datasets.
- **Complete code and notebook:** Can be executed to reproduce all figures and results.

7. Conclusion

This tutorial demonstrates the effective application of Vision Transformers (ViT) to the Fashion-MNIST dataset, highlighting a modular, reproducible, and accessible workflow. The ViT model achieved approximately 89% accuracy, which is competitive with traditional CNNs on this dataset. The use of global self-attention allows the model to extract robust features even from small grayscale images, capturing long-range dependencies that conventional convolutional approaches might miss. Despite its strong performance, challenges remain in distinguishing visually similar classes, indicating that techniques such as data augmentation or hybrid architectures combining CNNs and Transformers could further improve results. Overall, this work provides a professional and practical reference for students and practitioners aiming to implement Vision Transformers in PyTorch while ensuring reproducibility and clarity.

8. References

1. Dosovitskiy, A., et al. (2020). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. arXiv:2010.11929.
2. Vaswani, A., et al. (2017). *Attention is All You Need*. NeurIPS.
3. Xiao, H., Rasul, K., & Vollgraf, R. (2017). *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. arXiv:1708.07747.
4. PyTorch Documentation: https://pytorch.org/docs/stable/index.html
5. HuggingFace Transformers: https://huggingface.co/transformers/