Dimensionality Reduction using Autoencoders

An introduction to deep learning and data analysis using autoencoders for dimensionality reduction. Explore methods for simplifying complex datasets while preserving essential information.



What is Dimensionality Reduction?

Definition

Reduces high-dimensional data to lower dimensions.

Why do we need it?

- Faster processing
- Easier data visualization
- Noise reduction
- Better model performance

Examples

- Images
- Audio
- Genetic data

Traditional Methods: PCA

PCA (Principal Component Analysis)

Based on linear transformation.

Easy to interpret

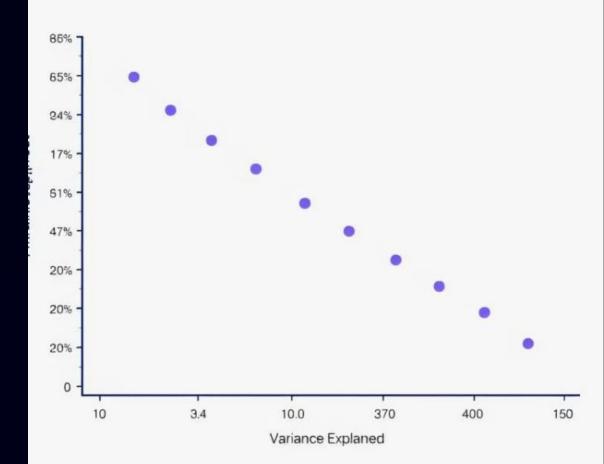
Limited with nonlinear data.



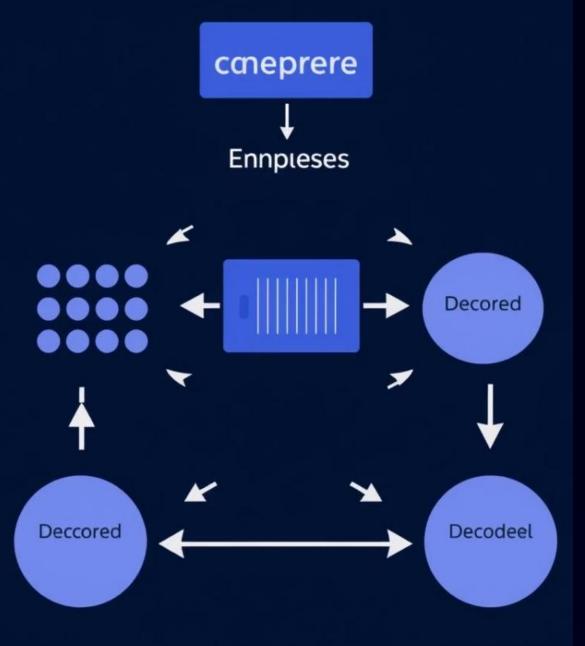
Reduces dimensions

Captures the maximum variance.

Principal Comonent 's Annet



Aluttocinceude



What is an Autoencoder?

Unsupervised neural network

Learns to compress data and reconstruct it.

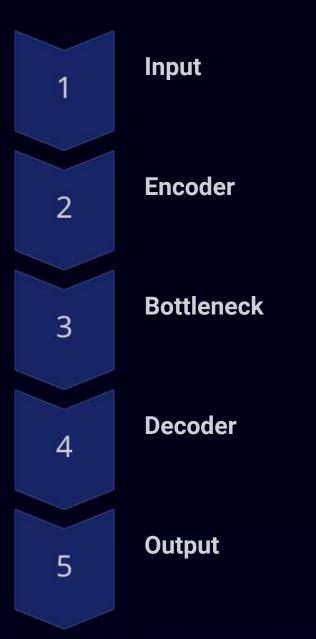
Encoder

Converts input to a lower dimension.

Decoder

Reconstructs input from the encoded data.

How Autoencoders Work



Goal: Make the output as close as possible to the original input. The "bottleneck" layer holds the compressed representation.

```
Purshistt: Compleutive aytlari
Micircationy Mahles lost

Tittalle, leve can Pranc-Trafte Pretion)

Streperte (addelbl).
   Yer Mode Nade(Prerfeiition)
        Disenteice)
        ("arvapl, Inchifiion: -tatp; Sae-(aefte profectation: New Footries)."
         Ipt: "9 Tep" (ome-Pyton-Flat scople pythons et arriveriation la minution les
         "Aatter-Tay (lentlg)
    We: lat Pythont: Cally loy)
                     Tercings water (onthuners doing (fatrontle) by (labe the time time the term
          Dat, Aprelerion, it lay)
         Lat, Nater"(lat Indalpemfactertiever, That Gasciffel)
          Deal, Insirnathe "ssig ent salt" concduid,
           ick: (inprfactionn lstigating, infeertfields and (until puttell)
            call: "Tartale wate ugnester tenters
   7ricall. Inst. Crohigo-ajo leg, SimbSin derstigtonl = Ton: Comp
           ofraule Tarteir-Ischioll-llest.iraditg:, tatom of lestition:
  Pictallt, Relassi, ntess[Poive nerdies], agplaty, New Island species
                  hefs-in last Protnessand, Pasile head lat hande la
   V7. Detarlthwoed (G9 low (Treatierarttion!)
```

Practical Example: Code

```
import numpy as np # type: ignore
import pandas as pd # type: ignore
import matplotlib.pyplot as plt # type: ignore
from sklearn.decomposition import PCA # type: ignore
from sklearn.preprocessing import MinMaxScaler # type: ignore
from tensorflow.keras.models import Model # type: ignore
from tensorflow.keras.layers import Input, Dense # type: ignore
from tensorflow.keras.optimizers import Adam # type: ignore
from tensorflow.keras.losses import MeanSquaredError # type: ignore
# 1. Load and preprocess diabetes data
df = pd.read csv('diabetes.csv')
x data = df.values
# Scale data to [0, 1] range
scaler = MinMaxScaler()
x data = scaler.fit transform(x data)
# Split into train/test (80/20)
split idx = int(0.8 * len(x data))
x train = x data[:split idx]
x test = x data[split idx:]
```

```
Purshistt: Compleutive aytlari
Whare:
Micirationy Mahles lost
Fittalle, leve can Pranc-Trafte Pretion)

Streperte (adlelb1).
                Yer Mode Nade(Prerfaiition)
                                              Disentelco)
                                             ("arvapl, Inchifition: -tatp; Sae-(asfte profectation: New Footsian)."
                                                 Ipt: "9 Tep" (ome-Pyton-Flat scople pythons et arriversation to mismiss be land
                                                    "Aatter-Tay (lentlg)
                       He: Lat Pythont: Cally loy)
                                                                                                                  Tercings water (onthuners doing, (Tetrontic) bay (Lake" them: Millian' Moor Millian' Millian' Moor Millian' M
                                                      Dat, Aprelerion, it lay)
                                                    Lat, Nater"(lat Indalpemfactertiever, That (seciffel)
                                                       Deal, Insirnathe "ssig ent salt" concduird,
                                                       (ick: (inprfactionn latigating, infertfletts and (annot putell)
                              eccedcall: "Tartale wate ugnester tenters
                    Vricall. Inst. Crohigo-ajo Jeg, SimbSin derstigioni - Your Comp
  Ne: Sntocanffition?

Ne: Sntocanffition?

Ne: Sofraule Tarteir-Ischiol-llest.iraulitg: tatom of Jestition:

Pictallt. Relassi, ntess[Foive nerdies], agplaty, how the present 
                                 tenof, -shefs-in last prochassand, Pasile heact let hands intended
                 V7. Detarlthword (69 low (Treatierarttion!)
```

Practical Code

```
# 2. Atoencoder Model
input dim = x train.shape[1]
encoding dim = 4 # Smaller bottleneck for tabular data
# Encoder
input layer = Input(shape=(input dim,))
encoded = Dense(16, activation='relu')(input layer)
encoded = Dense(8, activation='relu')(encoded)
encoded = Dense(encoding_dim, activation='relu')(encoded) # Bottleneck
# Decoder
decoded = Dense(8, activation='relu')(encoded)
decoded = Dense(16, activation='relu')(decoded)
decoded = Dense(input_dim, activation='sigmoid')(decoded)
# Full model
autoencoder = Model(input layer, decoded)
autoencoder.compile(optimizer=Adam(learning rate=0.001),
loss=MeanSquaredError())
```

```
Purshistt: Compleutive aytlari
Whare:

Micircationy Mahles lost

Fittalle. leve can Pranc-Traffe Pretion)

Losis Streperte (addelbl).
           Yer Mode Nade(Prerfaiition)
                            Disentelco)
                           ("arvaol, Inchifition: -tatp; Sae-(asfte profectation: War Footsine)."
                              Ipt: "9 Tep" (ome-Pyton-Flat scople pythons et arriveriation to minution benediction bened
                                "Aatter-Tay (lentlg)
               He: lat Pythont: Cally loy)
                                                                     Tercings water (onthuners doing ([atroulle] by (lake the distance has been been
                                 Dat, Aprelerion, it lay)
                                Lat, Nater"(lat Indalpemfactertiever, That (saciffel)
                                  Deal, Insirnathe "ssig ent salt" concduird,
                                   (ick: (inprfactionn lstigating, infertfleds and femot puted)
                  eccedcall: "Tartale wate ugnester tonters
             Writall. Inst. Crohigo-ajo Jeg, SimbSin derstigioni - Your Comp
              Ne: Sntocanffition?

Ne: Sofraule Tarteir-Ischioll-llest.irauitg:, tatom of lestition:
         Pictallt. Relassi, ntessipoive nerdiesi, agplaty, involument
  Pytenof, -shefs-in last prochassand, Pasile least let hands interest = incineliod
```

Practical Example: Code

```
# 3. Train the model
history = autoencoder.fit(x train, x train,
              epochs=100.
              batch size=32,
              shuffle=True,
              validation data=(x test, x test))
# 4. Reconstruct test data
reconstructed = autoencoder.predict(x test)
# 5. Compare with PCA
pca = PCA(n components=encoding dim)
pca.fit(x train)
x pca = pca.transform(x test)
x pca inverse = pca.inverse transform(x pca)
# 6. Calculate and compare MSE
def calculate mse(original, reconstructed, label):
  mse = np.mean(np.square(original - reconstructed))
  print(f"{label} MSE: {mse:.5f}")
calculate mse(x test, reconstructed, "Autoencoder")
calculate mse(x test, x pca inverse, "PCA")
```

```
Purshistt: Compleutive aytlari
Whare:

Micircationy Mahles lost

Fittable. leve can Pranc-Trafte Pretion)

Micircationy Mahles lost

Streperte (addelbl).
            Yer Mode Nade(Prerfaiition)
                             Disentelco)
                             ("arvapl, Inchifiion: -tatp; Sae-(aefte profectation: New Footries)."
                               Ipt: "9 Tep" (ome-Pyton-Flat scople pythons et arrierration to minute le land
                                "Aatter - Tay (lentlg)
                He: lat Pythont: Cally loy)
                                                                     Tercings water (onthuners doing (fatrontle) bay (lake the distance has been been
                                 Dat, Aprelerion, it lay)
                               Lat, Nater"(lat Indalpemfactertiever, That (saciffel)
                                  Deal, Insignathe "ssig ent salt" concduird,
                                   (ick: (inprfactionn latigating, infertfields and found putall)
                   escedcall: "Tartale wate ugnester tenters
              Vritall. Inst. Crohigp-ajo Jeg, RimbSin derstigioni - Your Comp
                              Sntocanffitiop?
Sntraule Tarteir-Ischioll-llest.iraditg: tatom of Jestitien:
            Pictallt. Relass, ntess(Poive nerddes), agplaty, Now Using our control of the con
                       tenof, -shefs-in losr prochassand, Pasile heact let hands lote
             V7. Detarlthwoed (G9 low (Treatierarttion!)
```

Practical Example: Code

```
# 7. Visualization - Plot first 5 features
plt.figure(figsize=(15, 10))
for feat in range(5): # Plot first 5 features
  plt.subplot(5, 1, feat+1)
  plt.plot(x_test[:50, feat], label='Original', marker='o')
  plt.plot(reconstructed[:50, feat], label='Autoencoder', marker='x')
  plt.plot(x pca inverse[:50, feat], label='PCA', marker='^')
  plt.title(f'Feature {feat+1} Comparison')
  plt.legend()
plt.tight layout()
plt.show()
# 8. Plot training history
plt.figure(figsize=(10, 5))
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val loss'], label='Validation Loss')
plt.title('Model Training History')
plt.ylabel('Loss (MSE)')
plt.xlabel('Epoch')
plt.legend()
plt.grid(True)
plt.show()
```

```
Pursfistf: Completive aytlort
Whare:
Micirationy Mahles lost
Tittalle, leve can Prame-Trafte Pretion)

Strengte (adlelb1).
          Ticvertable is Streperte (adle1b1).
    Yer Mode Nade(Prerfaition)
           Disentelco)
           (*arvaol, Inchifiion: -tatp; Sae-(asfle profectation: War Funtziant, N
           lpt: "9 Tep" (ome-Pyton-Flat scople pythons et acressiantes la martin le Suda
            "Aatter-Tay (lentlg)
     He: lat Pythont: Cally loy)
     Repaise (axt
                         Tercings water (onthuners doing ([atroutle) by (lake the thing was been been
      V). Dat, Aprelerion, it lay)
      V). Lat, Nater"(lat Indalpemfactertiever, That Gseciffel)
      V). Deal, Insirnathe "ssig ent salt" concduird,
      Av: (ick: (inprfactionn lstigating, infeertfields and femonically
      Pescedcall: "Tartale wate ugnester tenters
 Me: Sntocanffition?

Me: Sntraule Tarteir-Ischioli-llest.iraulits; tatom of Jestition:

Me: Sntraule Tarteir-Ischioli-nerdnesi, agplaty, find the common particular applaty.
V7. Detarlthwoed (69 low (Treatierarttion!)

Pytcnof, -shefs-in Insr Prochessand, Parile Heact let France

Net: =incineliod
```

output

```
PS COURSEYNUSS Set-Executionality -Executionality Sysses -Scope Process -Force
PS COURSEYNUS - Appendix prescription of the Course of the Cour
```

10

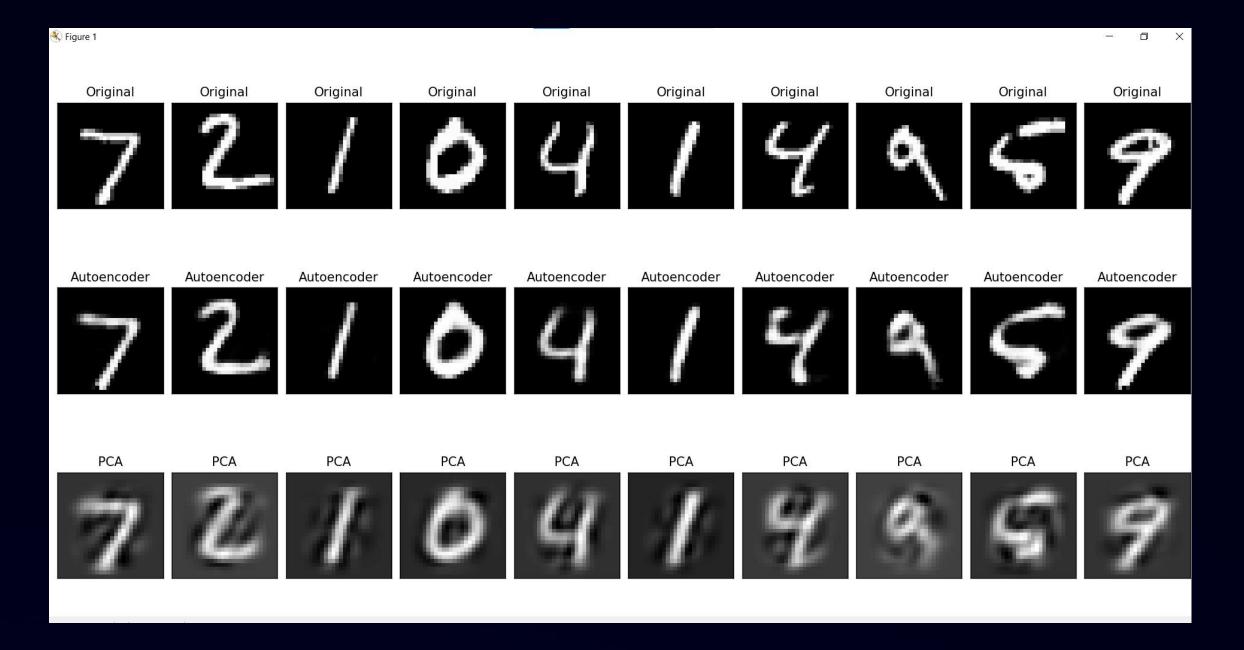
```
Purshistf: Compleutive sytlert
Micircationy Mahles lost
    Tittalle, leve can Pranc-Traffe Pretion)
     Ticvertable is Streperte (adle1b1).
Yer Mode Nade(Prerfaittion)
     Disenteice)
     ("arvapl, Inchiftion: -tatp; Sae-(asfte profectation: War Footside)."
      Ipt: "9 Tep" (ome-Pyton-Flat scople pythons et acreercolion 1) ome
      "Aatter - Tay (lentlg)
 He: lat Pythont: Cally loy)
                 Tercings water (onthuners doing ([atroutle) bay (labe" them; thing there has been
 V). Dat, Aprelerion, it lay)
 V). Lat. Nater"(lat Indalpemfactertiever, That Graciffel)
      Deal, Insirnathe "ssig ent salt" concduird,
 Av: (ick: (inprfactionn latigating, infeertfields and feming putter)
  Mescedcall: "Tartale wate ugnester tenters
 Vricall. Inst. Crohigo-ajo Jeg, SimbSin derstiptonl-: Tout Camp'
     Sofraule Tarteir-Ischioll-llest irauits: , tatom of lestition:
Pictallt. Relassi, ntess(Foive nerdites), agplaty, Now Ishat our commen
Pytenof, -shefs-in last Prothessand, Pasile feect let hands intended.
```

output

```
Epoch 22/50
                     ←[1m235/235←[0m ←[32m—
Epoch 23/50
<[1m235/235<[0m <[32m-
                      Epoch 24/50
←[1m235/235←[0m ←[32m-
                      +-[0m+[37m+[0m +[1m1s+[0m 5ms/step - loss: 0.0102 - val_loss: 0.0099
Epoch 25/50
←[1m235/235←[0m ←[32m-
                      Epoch 26/50
<[1m235/235<[0m <[32m-
                      Epoch 27/50
←[1m235/235←[0m ←[32m—
                      Epoch 28/50
←[1m235/235←[0m ←[32m-
                      -+[0m+[37m+[0m +[1m1s+[0m 5ms/step - loss: 0.0096 - val loss: 0.0093
Epoch 29/50
<[1m235/235<[0m <[32m-
                      Epoch 30/50
←[1m235/235←[0m ←[32m-
                      Epoch 31/50
←[1m235/235←[0m ←[32m-
                     Epoch 32/50
←[1m235/235←[0m ←[32m-
                      Epoch 33/50
+[1m235/235+[0m +[32m-
                      Epoch 34/50
←[1m235/235←[0m ←[32m—
                      -←[0m←[37m←[0m ←[1m1s←[0m 4ms/step - loss: 0.0090 - val loss: 0.0089
Epoch 35/50
←[1m235/235←[0m ←[32m-
                      Epoch 36/50
<[1m235/235<[0m <[32m-
                      Epoch 37/50
←[1m235/235←[0m ←[32m-
                      Epoch 38/50
←[1m235/235←[0m ←[32m-
                      +-[0m+[37m+[0m +[1m1s+[0m 4ms/step - loss: 0.0087 - val_loss: 0.0087
Epoch 39/50
←[1m235/235←[0m ←[32m—
                      Epoch 40/50
<[1m235/235<[0m <[32m-
                      -+[0m+[37m+[0m +[1m1s+[0m 5ms/step - loss: 0.0086 - val_loss: 0.0086
Epoch 41/50
←[1m235/235←[0m ←[32m-
                      +-[0m+[37m+[0m +[1m1s+[0m 6ms/step - loss: 0.0086 - val_loss: 0.0084
Epoch 42/50
←[1m235/235←[0m ←[32m-
                      Epoch 43/50
←[1m235/235←[0m ←[32m—
                      Epoch 44/50
←[1m235/235←[0m ←[32m—
                      --←[0m←[37m←[0m ←[1m1s←[0m 4ms/step - loss: 0.0084 - val_loss: 0.0084
Epoch 45/50
←[1m235/235←[0m ←[32m-
                      Epoch 46/50
<[1m235/235<[0m <[32m-
                      Epoch 47/50
←[1m235/235←[0m ←[32m-
                     —←[Am←[37m←[Am ←[1m1s←[Am 4ms/sten - loss・A AAR3 - val loss・A AAR3
```

11

final output



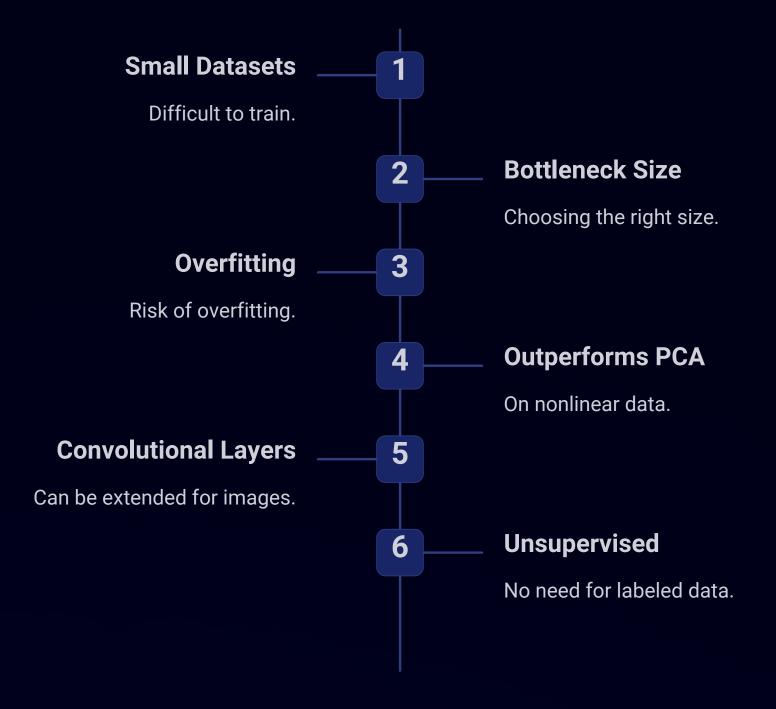
Autoencoder vs PCA Results

Original Images

Autoencoder Reconstruction

PCA Reconstruction

Challenges & Key Takeaways



Conclusion

Powerful Tool

For dimensionality reduction.

Preprocessing

Boosts other models performance.

Better Results

For complex data vs PCA.



The Power of Autoencoders

At the heart of dimensionality reduction lies an innovative technique - autoencoders. These neural networks possess a remarkable ability to uncover the underlying structure of complex data, allowing us to extract the most salient features while dramatically reducing the dimensionality. Autoencoders work by learning to encode the input data into a compact, low-dimensional representation, and then decode that representation back into the original input. Through this process, the network is able to identify the essential characteristics that define the data, discarding the superfluous details.

Unlike traditional methods like Principal Component Analysis (PCA), autoencoders are not limited by linear assumptions. They can capture intricate non-linear relationships, unlocking new possibilities for dimensionality reduction and feature extraction. This makes them a powerful tool for a wide range of applications, from image processing to natural language understanding.

By leveraging the power of autoencoders, we can gain deeper insights, streamline our workflows, and uncover hidden patterns that may have eluded us using conventional techniques. The key is to harness this innovative approach and unlock the true potential of your data. Further research autoencoders and their applications in real-world scenarios. Experiment with different architectures and datasets to gain a deeper understanding.

