Denoising Autoencoder Using Keras: A Practical Tutorial

1. Introduction:

In this modern era where data is most important and should be accurate, machine learning applications highly rely on the large volumes of image datasets under imperfect conditions. Real world images contain blur, noisy, sensor interference, and various distortions. Models trained directly on these noisy images often learn unstable patterns. One highly used neural network architecture that addresses this problem is the autoencoder. It's an unsupervised learning model which compresses input data and learns to reconstruct it.

A specifically influential autoencoder which is used thoroughly is Denoising Autoencoder (DAE), introduced by Vincent et al. (2008). The main concept is to corrupt the input with reasonable noise during training, meanwhile the network attempts to reconstruct the clean and original signal. This forces the model to learn noise invariant representations, robust that generalize beyond the corrupted data.

This tutorial demonstrates a clear and practical guide to training a denoising autoencoder by using the Keras deep learning library and Fashion MNIST dataset. The goal is to explain how DAE operates, how noise addition improves representation learning, and how effectively corrupted images can be reconstructed by using convolutional architecture.

2. Autoencoders:

Autoencoders are composed of two components, such as an encoder which maps an input x to a latent representation z and there's a decoder which reconstructs an approximation x ^. Bengio and Courville (2016) represent autoencoders as models which project data onto a lower dimensional manifold, securing essential structure and discarding noise.

- **Denoising Autoencoders:**

  Vincent et al (2008) proposed denoising autoencoder to put a stop to trivial copying and encourage robust feature learning. The encoder receives corrupted inputs $\tilde{x}$, instead of getting clean inputs, usually generated via masking or Gaussian noise. Such as the learning objective becomes:

  $$\min_{\theta} | \quad | x - f_{\theta}(\tilde{x}) | \quad |^2$$

  This enforces the autoencoder to map the noisy samples back to the data manifold, successfully performing as a local form of a regularisation. Bengio et al (2013) proclaim that DAE learns the structure of the data generating distribution by attracting the corrupted samples toward regions of high density.

3. Dataset Autoencoder architecture and Noise Model:

The dataset I used to build the model is Fashion-MINST, which contains 70,000 grayscale images representing ten categories of clothing. Each image consists of 28* 28

pixels. This dataset is highly used as a drop-in replacement for MNIST because of its richer texture and slightly higher difficulty.
All images are normalised to [0,1] before training and expanded to the shape of (28,28,1) for convolutional layers.

- **Noise model:**
  Using Vincent et al. (2008), to each input image Gaussian noise was added:

$$\tilde{x} = x + \epsilon, \quad \epsilon \sim N(0,0.5^2)$$

  Corruption generates grainy images that simulates realistic degradation:

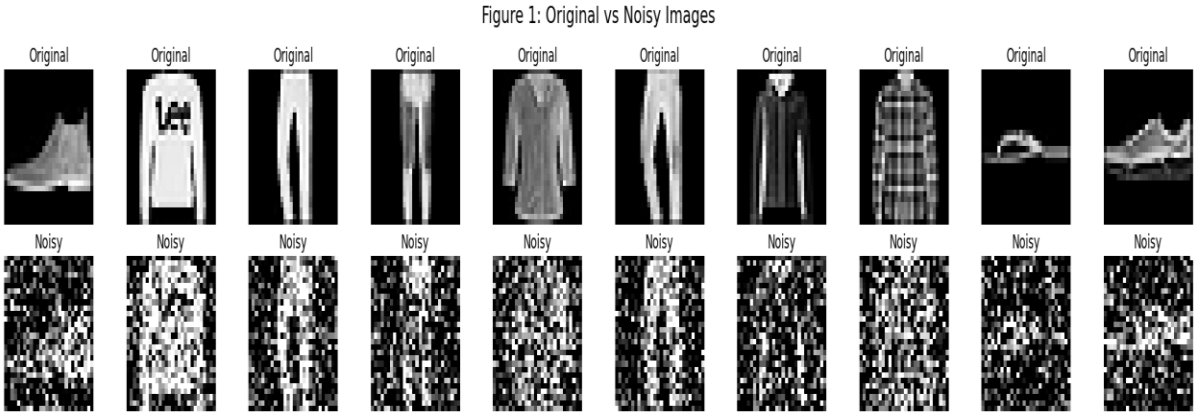

Figure 1: Top row is clean Fashion MNIST image and bottom row is their noisy version.

- **Model Architecture:**
  Due to the suitability of image data, A convolutional autoencoder is implemented, the encoder in this model compress the images to 7*7*64 tensor, then the decoder reconstructs the images back to their original size.
  Encoder structure:
  - Conv2D (32) → ReLU
  - Max Pooling (2D)
  - Conv2D (64) → ReLU
  - Max Pooling (2D)

  Decoder structure:

  - Conv2D (64)
  - Up Sampling 2D
  - Conv2D (32)
  - Up Sampling 2D
  - Conv2D (1, sigmoid)

The model is trained by using the Adam optimizer and for loss function Mean Squared Error (MSE) is used.

4. Implementation in Keras:

The following steps are implemented in the tutorial notebook:

- Loading Fashion MNIST dataset and normalizing pixel values.
- Inserting Gaussian noise to training and test images.
- Constructing encoder and decoder using Conv2d and UpSampling 2D.
- Training the model with the batch size of 128 and 20 epochs.
- Visualizing the results in the form of reconstruction quality and training curves.

Training is carried out on GPU runtime for efficiency.

5. Results:

The model is trained on 20 epochs.
Here is the key values extracted from the training outcome:

- Initial model training loss: 0.0430
- Initial model validation loss: 0.0205
- Final model training loss: 0.0140
- Final model validation loss: 0.0143

Interpretation:

The model training loss dropped from 0.0430 to 0.0140 indicating strong learning, and the model validation loss dropped from 0.0205 to 0.0143 to indicate good generalisation. There is notable overfitting the small gap between the losses to suggest that the model is well generalised.

This is specifically what we expect from a well-trained denoised autoencoder model.

Figure 2: Training vs Validation Loss

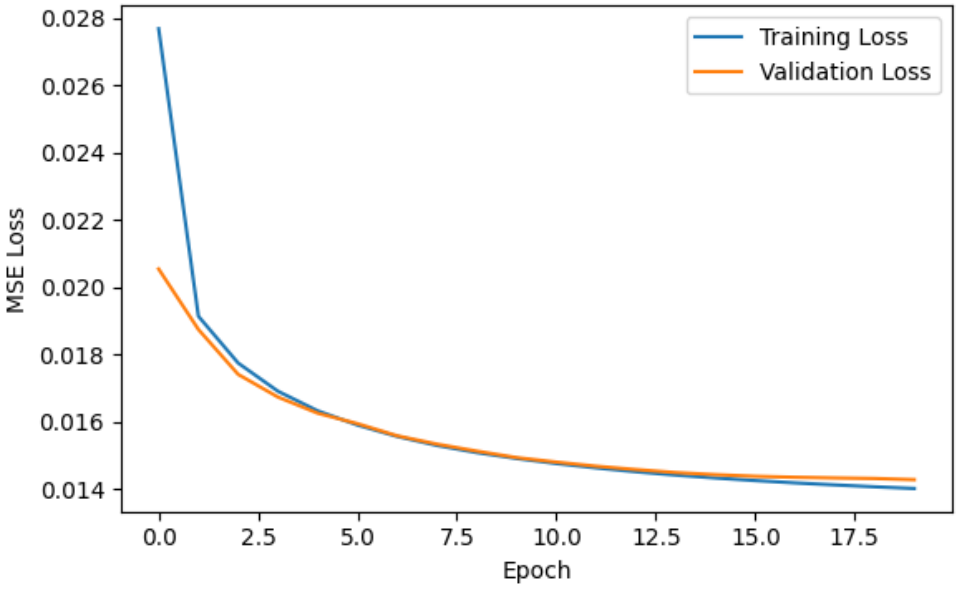

Figure 2: Training and validation MSE loss curve

- **Reconstruction quality:**
  To evaluate real world type data, the model is applied to noisy images. The output clearly shows the successful denoised outcomes. Across all categories, the model consistently removes random pixel noise and restores structure.
  Due to spatial smoothing of convolutional autoencoders, fine edges remain slightly blurred.

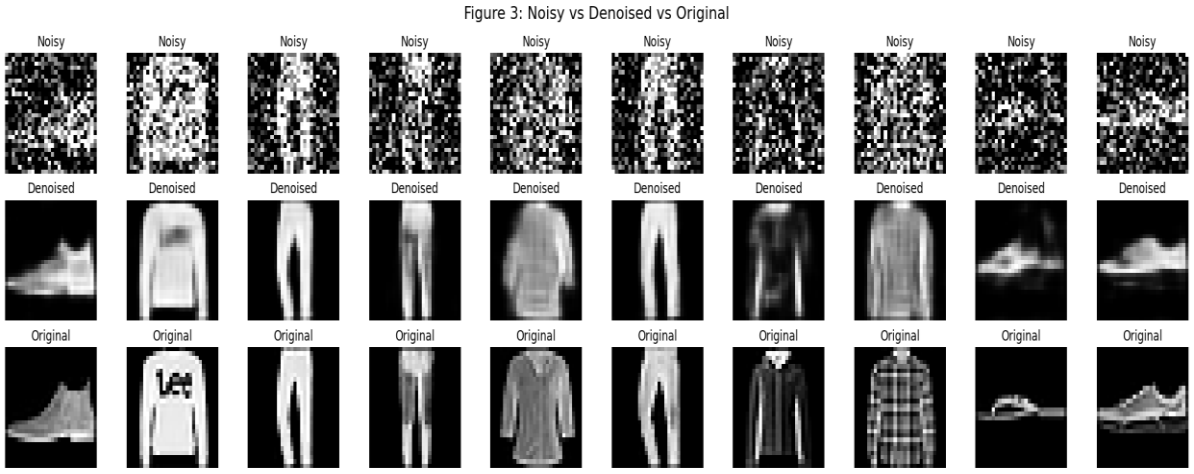

Figure 3: (Top row) Reconstruction comparison showing noisy images, (middle row) denoised outputs, and (bottom row) clean originals.

6. Discussion:

Our results which we got from training this model align very closely to the findings of Vincent et al. (2008), proving that DAEs learn robust, noise-invariant representations.

- **Strengths:**
  + Model shows strong generalisation (train 0.0140 vs test 0.0143).
  + Effectual noise removal.
  + Natural and smooth reconstruction of the images.
  + Stable trained model with no overfitting.
- **Limitations:**
  – Some fine textures are blurred, which is common for convolutional decoders.
  – All real-world distortions may not be represented in Gaussian noise.
  – Model performance depends strongly on selected noise factors.
- **Alignment with literature:**
  Bengio et al. (2013) argues that denoising autoencoders learn manifold structure through mapping noisy inputs toward high density regions. Our findings clearly support this behavior; images are continuously restored towards semantically meaningful shapes.

7. Ethical Consideration:

Autoencoders unintentionally memorise training images and can amplify dataset biases, or when used in safety critical contexts such as medical imaging can produce misleading reconstructions. So, for that responsible use requires:

- Awareness of all reconstruction failure cases.
- Clean and Diverse training data.
- Ensuring denoising does not destroy important details.
- Without confirming privacy, safeguards avoid using sensitive facial datasets.

8. Conclusion:

This tutorial illustrates how to implement a convolutional denoising autoencoder (DAE) using Keras and Fashion MNIST dataset. By using Gaussian noise, the model achieved robustness and reconstruct clean images by using corrupted inputs. The model training step was processed successfully and represents reconstruction quality quantitatively and visually strong.

For representational learning, noise removal and dimensionality reduction Denoising autoencoders remain a foundational tool, and they provide an accessible yet strong and powerful introduction to neural network based unsupervised learning.

9. Tutorials Github Link:

https://github.com/irzamlatif/denoising-autoencoder-tutorial.git

10. References:

- Heaton, J. (2017). Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning. *Genetic Programming and Evolvable Machines*, [online] 19(1-2), pp.305–307. doi:https://doi.org/10.1007/s10710-017-9314-z.
- Team, K. (2021). *Keras documentation: Convolutional autoencoder for image denoising*. [online] keras.io. Available at: https://keras.io/examples/vision/autoencoder/.
- TensorFlow. (n.d.). *Intro to Autoencoders | TensorFlow Core*. [online] Available at: https://www.tensorflow.org/tutorials/generative/autoencoder.
- Vincent, P., Larochelle, H., Bengio, Y. and Manzagol, P.-A. (2008). *Extracting and Composing Robust Features with Denoising Autoencoders*. [online] Available at: https://www.cs.toronto.edu/~larocheh/publications/icml-2008-denoising-autoencoders.pdf.
- Vincent, P., Ca, P., Larochelle, H., Toronto, L., Edu, Lajoie, I., Ca, Y. and Ca, P.-A. (2010). Stacked Denoising Autoencoders: Learning Useful Representations in a Deep

Network with a Local Denoising Criterion Yoshua Bengio Pierre-Antoine Manzagol. *Journal of Machine Learning Research*, [online] 11, pp.3371–3408. Available at: https://www.jmlr.org/papers/volume11/vincent10a/vincent10a.pdf.