Denoising Autoencoders (DAE)

Jérémie Cabessa

Laboratoire DAVID, UVSQ

Introduction

- **But d'un autoencodeur (rappel):** projeter les data dans un **espace latent (latent space)** et être capable de les reconstruire ces data.
- Applications en compression de data (data compression).
- Autre application des autoencdeurs: le **débruitage des data (denoising)**.
- Pour cela, on utilise des **denoising autoencoders**.

Introduction

- **But d'un autoencodeur (rappel):** projeter les data dans un **espace latent (latent space)** et être capable de les reconstruire ces data.
- Applications en compression de data (data compression).
- Autre application des autoencdeurs: le **débruitage des data (denoising)**.
- Pour cela, on utilise des **denoising autoencoders**.

Introduction

- **But d'un autoencodeur (rappel):** projeter les data dans un **espace latent (latent space)** et être capable de les reconstruire ces data.
- Applications en compression de data (data compression).
- Autre application des autoencdeurs: le **débruitage des data (denoising)**.
- Pour cela, on utilise des **denoising autoencoders**.

Introduction

- **But d'un autoencodeur (rappel):** projeter les data dans un **espace latent (latent space)** et être capable de les reconstruire ces data.
- Applications en compression de data (data compression).
- Autre application des autoencdeurs: le **débruitage des data (denoising)**.
- Pour cela, on utilise des **denoising autoencoders**.

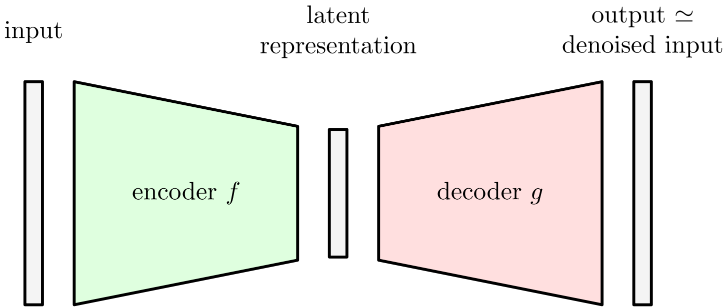
Introduction

- Principe similaire à celui d'un autoencodeur.
- Au lieu de viser la reconstruction des data, on vise leur débruitage.

Introduction

- Principe similaire à celui d'un autoencodeur.
- Au lieu de viser la reconstruction des data, on vise leur débruitage.

Architecture encodeur-décodeur



Architecture encodeur-décodeur

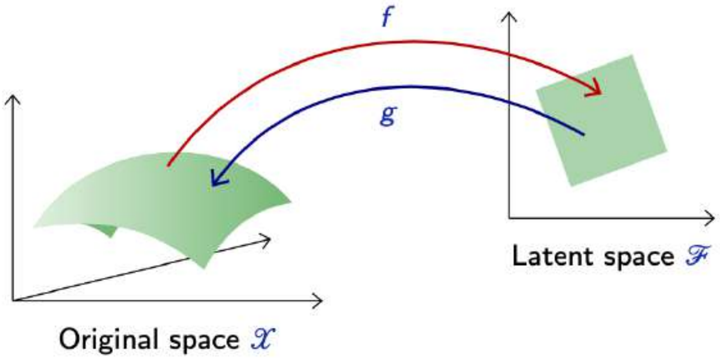


Figure taken from [Fleuret, 2022].

Denoising Autoencodeur (DAE)

- L'**encodeur** est un deep neural network $f(\cdot; \mathbf{\Theta_f})$.
- Le **décodeur** est un deep neural network $g(\cdot; \mathbf{\Theta_g})$.
- La composition de $f(\cdot; \mathbf{\Theta_f})$ et $g(\cdot; \mathbf{\Theta_g})$ forme un réseau de neurones $\mathcal{N}(\cdot; \mathbf{\Theta_f}, \mathbf{\Theta_g})$.
- $\mathcal{N}$ reçoit des inputs bruitées et son but est de les débruiter. Pour cela, on dispose de *data propres* comme *targets*.
- Si $\mathbf{x}$ et $\tilde{\mathbf{x}}$ représentent une data propre (clean) et bruitée (noisy), respectivement, où $\tilde{\mathbf{x}}$ est une perturbation de $\mathbf{x}$, alors on veut:

$$\mathcal{N}(\tilde{\mathbf{x}}; \mathbf{\Theta_f}, \mathbf{\Theta_g}) \quad = \quad g\left(f(\tilde{\mathbf{x}}; \mathbf{\Theta_f}); \mathbf{\Theta_g}\right) \quad \simeq \quad \mathbf{x}$$

Denoising Autoencodeur (DAE)

- L'**encodeur** est un deep neural network $f(\cdot; \mathbf{\Theta_f})$.
- Le **décodeur** est un deep neural network $g(\cdot; \mathbf{\Theta_g})$.
- La composition de $f(\cdot; \mathbf{\Theta_f})$ et $g(\cdot; \mathbf{\Theta_g})$ forme un réseau de neurones $\mathcal{N}(\cdot; \mathbf{\Theta_f}, \mathbf{\Theta_g})$.
- $\mathcal{N}$ reçoit des inputs bruitées et son but est de les débruiter. Pour cela, on dispose de *data propres* comme *targets*.
- Si $\mathbf{x}$ et $\tilde{\mathbf{x}}$ représentent une data propre (clean) et bruitée (noisy), respectivement, où $\tilde{\mathbf{x}}$ est une perturbation de $\mathbf{x}$, alors on veut:

$$\mathcal{N}(\tilde{\mathbf{x}}; \mathbf{\Theta_f}, \mathbf{\Theta_g}) = g\left(f(\tilde{\mathbf{x}}; \mathbf{\Theta_f}); \mathbf{\Theta_g}\right) \simeq \mathbf{x}$$

Denoising Autoencodeur (DAE)

- L'**encodeur** est un deep neural network $f(\cdot; \mathbf{\Theta_f})$.
- Le **décodeur** est un deep neural network $g(\cdot; \mathbf{\Theta_g})$.
- La composition de $f(\cdot; \mathbf{\Theta_f})$ et $g(\cdot; \mathbf{\Theta_g})$ forme un réseau de neurones $\mathcal{N}(\cdot; \mathbf{\Theta_f}, \mathbf{\Theta_g})$.
- $\mathcal{N}$ reçoit des inputs bruitées et son but est de les débruiter. Pour cela, on dispose de *data propres* comme *targets*.
- Si $\mathbf{x}$ et $\tilde{\mathbf{x}}$ représentent une data propre (clean) et bruitée (noisy), respectivement, où $\tilde{\mathbf{x}}$ est une perturbation de $\mathbf{x}$, alors on veut:

$$\mathcal{N}(\tilde{\mathbf{x}}; \mathbf{\Theta_f}, \mathbf{\Theta_g}) = g\left(f(\tilde{\mathbf{x}}; \mathbf{\Theta_f}); \mathbf{\Theta_g}\right) \simeq \mathbf{x}$$

Denoising Autoencodeur (DAE)

- L'**encodeur** est un deep neural network $f(\cdot; \mathbf{\Theta_f})$.
- Le **décodeur** est un deep neural network $g(\cdot; \mathbf{\Theta_g})$.
- La composition de $f(\cdot; \mathbf{\Theta_f})$ et $g(\cdot; \mathbf{\Theta_g})$ forme un réseau de neurones $\mathcal{N}(\cdot; \mathbf{\Theta_f}, \mathbf{\Theta_g})$.
- $\mathcal{N}$ reçoit des inputs bruitées et son but est de les débruiter. Pour cela, on dispose de *data propres* comme *targets*.
- Si $\mathbf{x}$ et $\tilde{\mathbf{x}}$ représentent une data propre (clean) et bruitée (noisy), respectivement, où $\tilde{\mathbf{x}}$ est une perturbation de $\mathbf{x}$, alors on veut:

$$\mathcal{N}(\tilde{\mathbf{x}}; \mathbf{\Theta_f}, \mathbf{\Theta_g}) = g\left(f(\tilde{\mathbf{x}}; \mathbf{\Theta_f}); \mathbf{\Theta_g}\right) \simeq \mathbf{x}$$

Denoising Autoencodeur (DAE)

- L'**encodeur** est un deep neural network $f(\cdot; \mathbf{\Theta_f})$.
- Le **décodeur** est un deep neural network $g(\cdot; \mathbf{\Theta_g})$.
- La composition de $f(\cdot; \mathbf{\Theta_f})$ et $g(\cdot; \mathbf{\Theta_g})$ forme un réseau de neurones $\mathcal{N}(\cdot; \mathbf{\Theta_f}, \mathbf{\Theta_g})$.
- $\mathcal{N}$ reçoit des inputs bruitées et son but est de les débruiter. Pour cela, on dispose de *data propres* comme *targets*.
- Si $\mathbf{x}$ et $\tilde{\mathbf{x}}$ représentent une data propre (clean) et bruitée (noisy), respectivement, où $\tilde{\mathbf{x}}$ est une perturbation de $\mathbf{x}$, alors on veut:

$$\mathcal{N}(\tilde{\mathbf{x}}; \mathbf{\Theta_f}, \mathbf{\Theta_g}) \quad = \quad g\left(f(\tilde{\mathbf{x}}; \mathbf{\Theta_f}); \mathbf{\Theta_g}\right) \quad \simeq \quad \mathbf{x}$$

Denoising Autoencodeur (DAE)

- Pour cela, on minimise la mean squared error (MSE):

$$\mathcal{L}(\mathbf{X}, \tilde{\mathbf{X}}; \mathbf{\Theta_f}, \mathbf{\Theta_g}) = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{x_i} - g\left(f(\tilde{\mathbf{x}}_\mathbf{i}; \mathbf{\Theta_f}); \mathbf{\Theta_g}\right)\|^2$$

  où $\mathbf{X}$ et $\tilde{\mathbf{X}}$ représente la concaténation des data propre et bruitées, respectivement.

- Un exemple de bruitage serait:

$$\tilde{\mathbf{x}}_\mathbf{i} = \mathbf{x_i} + \boldsymbol{\epsilon}_\mathbf{i} \quad \text{où} \quad \boldsymbol{\epsilon}_\mathbf{i} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

- L'entraînement du réseau consiste alors à trouver les poids $\hat{\mathbf{\Theta}}_\mathbf{f}$ et $\hat{\mathbf{\Theta}}_\mathbf{g}$ qui satisfont

$$\hat{\mathbf{\Theta}}_\mathbf{f}; \hat{\mathbf{\Theta}}_\mathbf{g} = \arg\min_{\mathbf{\Theta_f}; \mathbf{\Theta_g}} \mathcal{L}(\mathbf{X}, \tilde{\mathbf{X}}; \mathbf{\Theta_f}, \mathbf{\Theta_g})$$

Denoising Autoencodeur (DAE)

- Pour cela, on minimise la mean squared error (MSE):

$$\mathcal{L}(\mathbf{X}, \tilde{\mathbf{X}}; \boldsymbol{\Theta}_\mathbf{f}, \boldsymbol{\Theta}_\mathbf{g}) = \frac{1}{N} \sum_{i=1}^{N} \| \mathbf{x}_\mathbf{i} - g\left(f(\tilde{\mathbf{x}}_\mathbf{i}; \boldsymbol{\Theta}_\mathbf{f}); \boldsymbol{\Theta}_\mathbf{g}\right) \|^2$$

  où $\mathbf{X}$ et $\tilde{\mathbf{X}}$ représente la concaténation des data propre et bruitées, respectivement.

- Un exemple de bruitage serait:

$$\tilde{\mathbf{x}}_\mathbf{i} = \mathbf{x}_\mathbf{i} + \boldsymbol{\epsilon}_\mathbf{i} \quad \text{où} \quad \boldsymbol{\epsilon}_\mathbf{i} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

- L'entraînement du réseau consiste alors à trouver les poids $\hat{\Theta}_\mathbf{f}$ et $\hat{\Theta}_\mathbf{g}$ qui satisfont

$$\hat{\Theta}_\mathbf{f}; \hat{\Theta}_\mathbf{g} = \underset{\Theta_\mathbf{f}; \Theta_\mathbf{g}}{\arg\min} \mathcal{L}(\mathbf{X}, \tilde{\mathbf{X}}; \Theta_\mathbf{f}, \Theta_\mathbf{g})$$

Denoising Autoencodeur (DAE)

- Pour cela, on minimise la mean squared error (MSE):

$$\mathcal{L}(\mathbf{X}, \tilde{\mathbf{X}}; \mathbf{\Theta_f}, \mathbf{\Theta_g}) = \frac{1}{N} \sum_{i=1}^{N} \| \mathbf{x_i} - g\left(f(\tilde{\mathbf{x}}_\mathbf{i}; \mathbf{\Theta_f}); \mathbf{\Theta_g}\right) \|^2$$

  où $\mathbf{X}$ et $\tilde{\mathbf{X}}$ représente la concaténation des data propre et bruitées, respectivement.

- Un exemple de bruitage serait:

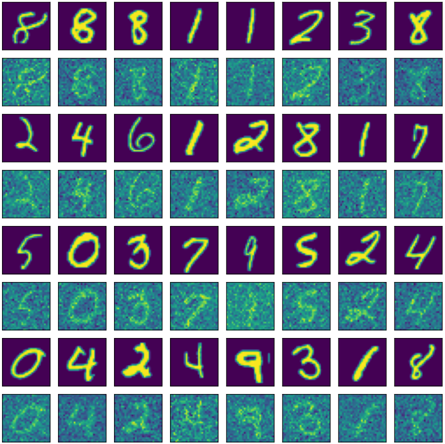
$$\tilde{\mathbf{x}}_\mathbf{i} = \mathbf{x_i} + \boldsymbol{\epsilon}_\mathbf{i} \quad \text{où} \quad \boldsymbol{\epsilon}_\mathbf{i} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

- L'entraînement du réseau consiste alors à trouver les poids $\hat{\mathbf{\Theta}}_\mathbf{f}$ et $\hat{\mathbf{\Theta}}_\mathbf{g}$ qui satisfont

$$\hat{\mathbf{\Theta}}_\mathbf{f}; \hat{\mathbf{\Theta}}_\mathbf{g} = \underset{\mathbf{\Theta_f}; \mathbf{\Theta_g}}{\arg\min}\, \mathcal{L}(\mathbf{X}, \tilde{\mathbf{X}}; \mathbf{\Theta_f}, \mathbf{\Theta_g})$$

Implémentation: DAE convolutionnel

- Data bruitées passées au DAE.



Implémentation: DAE convolutionnel

```
class DenoisingAutoencoder_CNN(nn.Module):
    """Implements a CNN denoising autoencoder"""

    def __init__(self):
        """constructor"""

        super().__init__()

        # N, 1, 28, 28
        self.encoder = nn.Sequential(
            # -> N, 16, 14, 14
            nn.Conv2d(1, 16, 3, stride=2, padding=1),
            nn.ReLU(),
            # -> N, 32, 7, 7
            nn.Conv2d(16, 32, 3, stride=2, padding=1),
            nn.ReLU(),
            # -> N, 64, 1, 1
            nn.Conv2d(32, 64, 7)
            )

        # N , 64, 1, 1
        self.decoder = nn.Sequential(
            # -> N, 32, 7, 7
            nn.ConvTranspose2d(64, 32, 7),
            nn.ReLU(),
            # N, 16, 14, 14 (N, 16, 13, 13 without output_padding)
            nn.ConvTranspose2d(32, 16, 3,
                               stride=2, padding=1, output_padding=1),
            nn.ReLU(),
            # N, 1, 28, 28 (N,1, 27, 27)
            nn.ConvTranspose2d(16, 1, 3,
                               stride=2, padding=1, output_padding=1),
            nn.Sigmoid()
            )

    def forward(self, x):
        """forward function"""

        encoded_data = self.encoder(x)
        decoded_data = self.decoder(encoded_data)

        return decoded_data
```

Implémentation: DAE convolutionnel

```
self.encoder = nn.Sequential(
    nn.Linear(28 * 28, 128),
    nn.ReLU(),
    nn.Linear(128, 64),
    nn.ReLU(),
    nn.Linear(64, 12),
    nn.ReLU(),
    nn.Linear(12, 2) # -> N, 2 only!
)
```

Implémentation: DAE convolutionnel

```
self.decoder = nn.Sequential(
    nn.Linear(2, 12),
    nn.ReLU(),
    nn.Linear(12, 64),
    nn.ReLU(),
    nn.Linear(64, 128),
    nn.ReLU(),
    nn.Linear(128, 28 * 28),
    nn.Sigmoid()
)
```

Implémentation: DAE convolutionnel

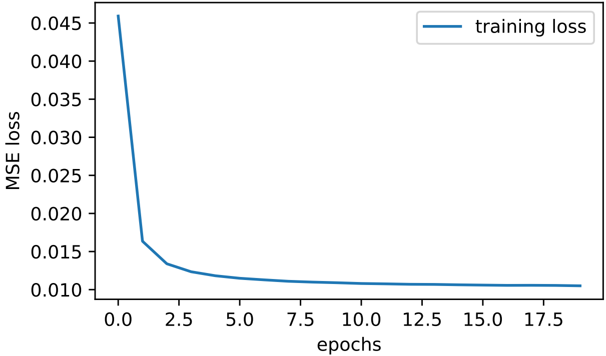
```
model = DenoisingAutoencoder_CNN()
criterion = nn.MSELoss()
optimizer = torch.optim.Adam(model.parameters(),
                             lr=1e-3,
                             weight_decay=1e-5)
```

Implémentation: DAE convolutionnel

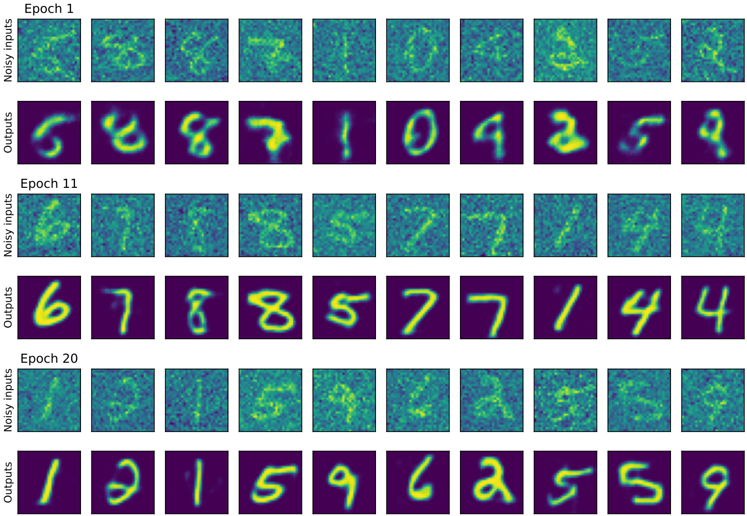
- **Training:** on calcule la MSE loss entre les images des data bruitées (images of noisy data) et les data propres (clean data, targets!).

```
for (inputs, _) in train_loader:      # clean inputs              (1)

    inputs_noisy = add_noise(inputs)  # noisy inputs
    outputs = model(inputs_noisy)     # images of noisy inputs (2)

    # loss between images of noisy inputs and clean inputs
    loss = criterion(outputs, inputs) # loss between (2) and (1)
    train_loss.append(loss.item())

    optimizer.zero_grad()
    loss.backward()
    optimizer.step()
```

Implémentation: DAE convolutionnel



Implémentation: DAE convolutionnel



Remarques

- Au fur et à mesure de l'entraînement, le DAE débruite les data de mieux en mieux.
- Le DAE est capable d'apprendre à débruiter les data.

Remarques

- Au fur et à mesure de l'entraînement, le DAE débruite les data de mieux en mieux.
- Le DAE est capable d'apprendre à débruiter les data.

Paradigme Noise2Noise

- Jusqu'ici, on a appris à débruiter des data à partir de data propres (loss between images of noisy data and clean data).
- Mais il se peut que l'on ne dispose pas de data propres (clean data) comme targets pour entraîner notre DAE.
- De manière surprenante, on peut apprendre à débruiter des data à partir de data elles-mêmes bruitées!
- C'est le paradigme "**noise to noise**".

Paradigme Noise2Noise

- Jusqu'ici, on a appris à débruiter des data à partir de data propres (loss between images of noisy data and clean data).
- Mais il se peut que l'on ne dispose pas de data propres (clean data) comme targets pour entraîner notre DAE.
- De manière surprenante, on peut apprendre à débruiter des data à partir de data elles-mêmes bruitées!
- C'est le paradigme "**noise to noise**".

Paradigme Noise2Noise

- Jusqu'ici, on a appris à débruiter des data à partir de data propres (loss between images of noisy data and clean data).
- Mais il se peut que l'on ne dispose pas de data propres (clean data) comme targets pour entraîner notre DAE.
- De manière surprenante, on peut apprendre à débruiter des data à partir de data elles-mêmes bruitées!
- C'est le paradigme "**noise to noise**".

Paradigme Noise2Noise

- Jusqu'ici, on a appris à débruiter des data à partir de data propres (loss between images of noisy data and clean data).
- Mais il se peut que l'on ne dispose pas de data propres (clean data) comme targets pour entraîner notre DAE.
- De manière surprenante, on peut apprendre à débruiter des data à partir de data elles-mêmes bruitées!
- C'est le paradigme "**noise to noise**".

Denoising Autoencodeur (DAE)

- L'**encodeur** est un deep neural network $f(\cdot; \mathbf{\Theta_f})$.
- Le **décodeur** est un deep neural network $g(\cdot; \mathbf{\Theta_g})$.
- La composition de $f(\cdot; \mathbf{\Theta_f})$ et $g(\cdot; \mathbf{\Theta_g})$ forme un réseau de neurones $\mathcal{N}(\cdot; \mathbf{\Theta_f}, \mathbf{\Theta_g})$.
- Une fois encore, $\mathcal{N}$ reçoit des inputs bruitées et son but est de les débruiter. Mais dans ce cas, on ne dispose pas de data propres, mais d'*autres data bruitées* comme *targets*.
- Si $\bar{\mathbf{x}}$ et $\tilde{\mathbf{x}}$ représentent deux versions bruitées indépendantes d'une même data propre inconnue $\mathbf{x}$, alors on a:

$$\mathcal{N}(\tilde{\mathbf{x}}; \mathbf{\Theta_f}, \mathbf{\Theta_g}) \quad = \quad g\left(f(\tilde{\mathbf{x}}; \mathbf{\Theta_f}); \mathbf{\Theta_g}\right) \quad \simeq \quad \bar{\mathbf{x}}$$

Denoising Autoencodeur (DAE)

- L'**encodeur** est un deep neural network $f(\cdot; \mathbf{\Theta_f})$.
- Le **décodeur** est un deep neural network $g(\cdot; \mathbf{\Theta_g})$.
- La composition de $f(\cdot; \mathbf{\Theta_f})$ et $g(\cdot; \mathbf{\Theta_g})$ forme un réseau de neurones $\mathcal{N}(\cdot; \mathbf{\Theta_f}, \mathbf{\Theta_g})$.
- Une fois encore, $\mathcal{N}$ reçoit des inputs bruitées et son but est de les débruiter. Mais dans ce cas, on ne dispose pas de data propres, mais d'*autres data bruitées* comme *targets*.
- Si $\bar{\mathrm{x}}$ et $\tilde{\mathrm{x}}$ représentent deux versions bruitées indépendantes d'une même data propre inconnue x, alors on a:

$$\mathcal{N}(\tilde{\mathrm{x}}; \mathbf{\Theta_f}, \mathbf{\Theta_g}) = g\left(f(\tilde{\mathrm{x}}; \mathbf{\Theta_f}); \mathbf{\Theta_g}\right) \simeq \bar{\mathrm{x}}$$

Denoising Autoencodeur (DAE)

- L'**encodeur** est un deep neural network $f(\cdot;\boldsymbol{\Theta}_\mathbf{f})$.
- Le **décodeur** est un deep neural network $g(\cdot;\boldsymbol{\Theta}_\mathbf{g})$.
- La composition de $f(\cdot;\boldsymbol{\Theta}_\mathbf{f})$ et $g(\cdot;\boldsymbol{\Theta}_\mathbf{g})$ forme un réseau de neurones $\mathcal{N}(\cdot;\boldsymbol{\Theta}_\mathbf{f},\boldsymbol{\Theta}_\mathbf{g})$.
- Une fois encore, $\mathcal{N}$ reçoit des inputs bruitées et son but est de les débruiter. Mais dans ce cas, on ne dispose pas de data propres, mais d'*autres data bruitées* comme *targets*.
- Si $\bar{\mathrm{x}}$ et $\tilde{\mathrm{x}}$ représentent deux versions bruitées indépendantes d'une même data propre inconnue x, alors on a:

$$\mathcal{N}(\tilde{\mathrm{x}};\boldsymbol{\Theta}_\mathbf{f},\boldsymbol{\Theta}_\mathbf{g}) \quad = \quad g\left(f(\tilde{\mathrm{x}};\boldsymbol{\Theta}_\mathbf{f});\boldsymbol{\Theta}_\mathbf{g}\right) \quad \simeq \quad \bar{\mathrm{x}}$$

Denoising Autoencodeur (DAE)

- L'**encodeur** est un deep neural network $f(\cdot; \mathbf{\Theta_f})$.
- Le **décodeur** est un deep neural network $g(\cdot; \mathbf{\Theta_g})$.
- La composition de $f(\cdot; \mathbf{\Theta_f})$ et $g(\cdot; \mathbf{\Theta_g})$ forme un réseau de neurones $\mathcal{N}(\cdot; \mathbf{\Theta_f}, \mathbf{\Theta_g})$.
- Une fois encore, $\mathcal{N}$ reçoit des inputs bruitées et son but est de les débruiter. Mais dans ce cas, on ne dispose pas de data propres, mais d'*autres data bruitées* comme *targets*.
- Si $\bar{\mathrm{x}}$ et $\tilde{\mathrm{x}}$ représentent deux versions bruitées indépendantes d'une même data propre inconnue x, alors on a:

$$\mathcal{N}(\tilde{\mathrm{x}}; \mathbf{\Theta_f}, \mathbf{\Theta_g}) \;=\; g\left(f(\tilde{\mathrm{x}}; \mathbf{\Theta_f}); \mathbf{\Theta_g}\right) \;\simeq\; \bar{\mathrm{x}}$$

Denoising Autoencodeur (DAE)

- L'**encodeur** est un deep neural network $f(\cdot; \mathbf{\Theta_f})$.
- Le **décodeur** est un deep neural network $g(\cdot; \mathbf{\Theta_g})$.
- La composition de $f(\cdot; \mathbf{\Theta_f})$ et $g(\cdot; \mathbf{\Theta_g})$ forme un réseau de neurones $\mathcal{N}(\cdot; \mathbf{\Theta_f}, \mathbf{\Theta_g})$.
- Une fois encore, $\mathcal{N}$ reçoit des inputs bruitées et son but est de les débruiter. Mais dans ce cas, on ne dispose pas de data propres, mais d'*autres data bruitées* comme *targets*.
- Si $\bar{\mathbf{x}}$ et $\tilde{\mathbf{x}}$ représentent deux versions bruitées indépendantes d'une même data propre inconnue $\mathbf{x}$, alors on a:

$$\mathcal{N}(\tilde{\mathbf{x}}; \mathbf{\Theta_f}, \mathbf{\Theta_g}) \quad = \quad g\left(f(\tilde{\mathbf{x}}; \mathbf{\Theta_f}); \mathbf{\Theta_g}\right) \quad \simeq \quad \bar{\mathbf{x}}$$

Denoising Autoencodeur (DAE)

- Pour cela, on minimise la mean squared error (MSE):

$$\mathcal{L}(\bar{\mathbf{X}}, \tilde{\mathbf{X}}; \boldsymbol{\Theta}_\mathbf{f}, \boldsymbol{\Theta}_\mathbf{g}) = \frac{1}{N} \sum_{i=1}^{N} \| \bar{\mathbf{x}}_\mathbf{i} - g\left(f(\tilde{\mathbf{x}}_\mathbf{i}; \boldsymbol{\Theta}_\mathbf{f}); \boldsymbol{\Theta}_\mathbf{g}\right) \|^2$$

  où $\bar{\mathbf{X}}$ et $\tilde{\mathbf{X}}$ représente la concaténation des deux types de data bruitées, respectivement.

- Un exemple de bruitage serait:

$$\begin{aligned} \bar{\mathbf{x}}_\mathbf{i} &= \mathbf{x}_\mathbf{i} + \boldsymbol{\epsilon}_\mathbf{i} \quad \text{où} \quad \boldsymbol{\epsilon}_\mathbf{i} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ \tilde{\mathbf{x}}_\mathbf{i} &= \mathbf{x}_\mathbf{i} + \boldsymbol{\epsilon}'_\mathbf{i} \quad \text{où} \quad \boldsymbol{\epsilon}'_\mathbf{i} \sim \mathcal{N}(\boldsymbol{\mu}', \boldsymbol{\Sigma}') \end{aligned}$$

- L'entraînement du réseau consiste alors à trouver les poids $\hat{\Theta}_\mathbf{f}$ et $\hat{\Theta}_\mathbf{g}$ qui satisfont

$$\hat{\Theta}_\mathbf{f}; \hat{\Theta}_\mathbf{g} = \arg\min_{\Theta_\mathbf{f}; \Theta_\mathbf{g}} \mathcal{L}(\bar{\mathbf{X}}, \tilde{\mathbf{X}}; \Theta_\mathbf{f}, \Theta_\mathbf{g})$$

Denoising Autoencodeur (DAE)

- Pour cela, on minimise la mean squared error (MSE):

$$\mathcal{L}(\bar{\mathbf{X}}, \tilde{\mathbf{X}}; \mathbf{\Theta_f}, \mathbf{\Theta_g}) = \frac{1}{N} \sum_{i=1}^{N} \left\| \bar{\mathbf{x}}_\mathbf{i} - g\left(f(\tilde{\mathbf{x}}_\mathbf{i}; \mathbf{\Theta_f}); \mathbf{\Theta_g}\right) \right\|^2$$

  où $\bar{\mathbf{X}}$ et $\tilde{\mathbf{X}}$ représente la concaténation des deux types de data bruitées, respectivement.

- Un exemple de bruitage serait:

$$\begin{aligned} \bar{\mathbf{x}}_\mathbf{i} &= \mathbf{x}_\mathbf{i} + \boldsymbol{\epsilon}_\mathbf{i} \quad \text{où} \quad \boldsymbol{\epsilon}_\mathbf{i} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ \tilde{\mathbf{x}}_\mathbf{i} &= \mathbf{x}_\mathbf{i} + \boldsymbol{\epsilon}'_\mathbf{i} \quad \text{où} \quad \boldsymbol{\epsilon}'_\mathbf{i} \sim \mathcal{N}(\boldsymbol{\mu}', \boldsymbol{\Sigma}') \end{aligned}$$

- L'entraînement du réseau consiste alors à trouver les poids $\hat{\Theta}_\mathbf{f}$ et $\hat{\Theta}_\mathbf{g}$ qui satisfont

$$\hat{\Theta}_\mathbf{f}; \hat{\Theta}_\mathbf{g} = \underset{\Theta_\mathbf{f}; \Theta_\mathbf{g}}{\arg\min}\, \mathcal{L}(\bar{\mathbf{X}}, \tilde{\mathbf{X}}; \Theta_\mathbf{f}, \Theta_\mathbf{g})$$

Denoising Autoencodeur (DAE)

- Pour cela, on minimise la mean squared error (MSE):

$$\mathcal{L}(\bar{\mathbf{X}}, \tilde{\mathbf{X}}; \mathbf{\Theta_f}, \mathbf{\Theta_g}) = \frac{1}{N} \sum_{i=1}^{N} \| \bar{\mathbf{x}}_\mathbf{i} - g\left(f(\tilde{\mathbf{x}}_\mathbf{i}; \mathbf{\Theta_f}); \mathbf{\Theta_g}\right) \|^2$$

  où $\bar{\mathbf{X}}$ et $\tilde{\mathbf{X}}$ représente la concaténation des deux types de data bruitées, respectivement.

- Un exemple de bruitage serait:

$$\begin{aligned} \bar{\mathbf{x}}_\mathbf{i} &= \mathbf{x_i} + \boldsymbol{\epsilon}_\mathbf{i} \quad \text{où} \quad \boldsymbol{\epsilon}_\mathbf{i} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ \tilde{\mathbf{x}}_\mathbf{i} &= \mathbf{x_i} + \boldsymbol{\epsilon}'_\mathbf{i} \quad \text{où} \quad \boldsymbol{\epsilon}'_\mathbf{i} \sim \mathcal{N}(\boldsymbol{\mu}', \boldsymbol{\Sigma}') \end{aligned}$$

- L'entraînement du réseau consiste alors à trouver les poids $\hat{\mathbf{\Theta}}_\mathbf{f}$ et $\hat{\mathbf{\Theta}}_\mathbf{g}$ qui satisfont

$$\hat{\mathbf{\Theta}}_\mathbf{f}; \hat{\mathbf{\Theta}}_\mathbf{g} = \underset{\mathbf{\Theta_f}; \mathbf{\Theta_g}}{\arg\min}\, \mathcal{L}(\bar{\mathbf{X}}, \tilde{\mathbf{X}}; \mathbf{\Theta_f}, \mathbf{\Theta_g})$$

Implémentation: DAE convolutionnel

```python
class DenoisingAutoencoder_CNN(nn.Module):
    """Implements a CNN denoising autoencoder"""

    def __init__(self):
        """constructor"""

        super().__init__()

        # N, 1, 28, 28
        self.encoder = nn.Sequential(
            # -> N, 16, 14, 14
            nn.Conv2d(1, 16, 3, stride=2, padding=1),
            nn.ReLU(),
            # -> N, 32, 7, 7
            nn.Conv2d(16, 32, 3, stride=2, padding=1),
            nn.ReLU(),
            # -> N, 64, 1, 1
            nn.Conv2d(32, 64, 7)
            )

        # N , 64, 1, 1
        self.decoder = nn.Sequential(
            # -> N, 32, 7, 7
            nn.ConvTranspose2d(64, 32, 7),
            nn.ReLU(),
            # N, 16, 14, 14 (N, 16, 13, 13 without output_padding)
            nn.ConvTranspose2d(32, 16, 3,
                               stride=2, padding=1, output_padding=1),
            nn.ReLU(),
            # N, 1, 28, 28 (N,1, 27, 27)
            nn.ConvTranspose2d(16, 1, 3,
                               stride=2, padding=1, output_padding=1),
            nn.Sigmoid()
            )

    def forward(self, x):
        """forward function"""

        encoded_data = self.encoder(x)
        decoded_data = self.decoder(encoded_data)

        return decoded_data
```

Implémentation: DAE convolutionnel

```
self.encoder = nn.Sequential(
    nn.Linear(28 * 28, 128),
    nn.ReLU(),
    nn.Linear(128, 64),
    nn.ReLU(),
    nn.Linear(64, 12),
    nn.ReLU(),
    nn.Linear(12, 2) # -> N, 2 only!
)
```

Implémentation: DAE convolutionnel

```
self.decoder = nn.Sequential(
    nn.Linear(2, 12),
    nn.ReLU(),
    nn.Linear(12, 64),
    nn.ReLU(),
    nn.Linear(64, 128),
    nn.ReLU(),
    nn.Linear(128, 28 * 28),
    nn.Sigmoid()
)
```

Implémentation: DAE convolutionnel

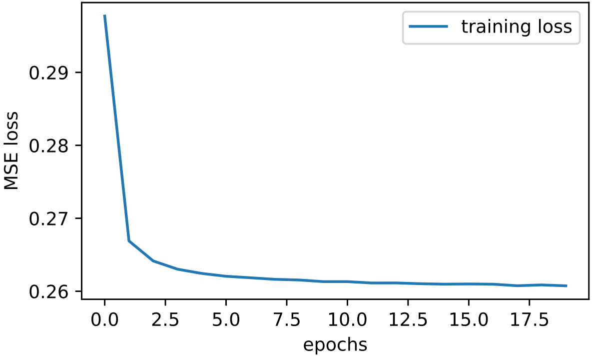
```
model = DenoisingAutoencoder_CNN()
criterion = nn.MSELoss()
optimizer = torch.optim.Adam(model.parameters(),
                             lr=1e-3,
                             weight_decay=1e-5)
```

Implémentation: DAE convolutionnel

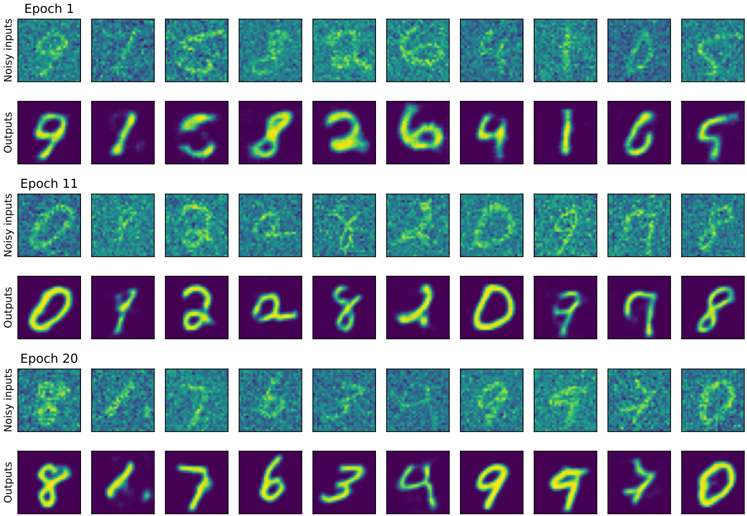
- **Training**: on calcule la MSE loss les images des data bruitées (images of noisy data) et les data bruitées initiales (noisy data, targets!).

```
for (inputs, _) in train_loader:

    inputs_noisy1 = add_noise(inputs) # noisy inputs!        (1)
    inputs_noisy2 = add_noise(inputs) # other noisy inputs
    outputs = model(inputs_noisy2)    # images of noisy inputs (2)

    # loss between noisy inputs and images of other noisy inputs
    loss = criterion(outputs, inputs_noisy1) # loss beween (2) and (1)
    train_loss.append(loss.item())

    optimizer.zero_grad()
    loss.backward()
    optimizer.step()
```

Implémentation: DAE convolutionnel



Implémentation: DAE convolutionnel



Remarques

- Bien que le DAE soit entraîné sur des data bruitées uniquement, au fur et à mesure de son entraînement, il apprend à débruiter ces data de mieux en mieux.
- Le DAE est capable de débruiter les data, même sans avoir été entraîné sur des data propres!
- Ceci n'a rien de magique, et nous allons le montrer dans les slides suivants.
- L'hypothèse d'indépendance des deux bruits est cruciale.

Remarques

- Bien que le DAE soit entraîné sur des data bruitées uniquement, au fur et à mesure de son entraînement, il apprend à débruiter ces data de mieux en mieux.
- Le DAE est capable de débruiter les data, même sans avoir été entraîné sur des data propres!
- Ceci n'a rien de magique, et nous allons le montrer dans les slides suivants.
- L'hypothèse d'indépendance des deux bruits est cruciale.

Remarques

- Bien que le DAE soit entraîné sur des data bruitées uniquement, au fur et à mesure de son entraînement, il apprend à débruiter ces data de mieux en mieux.
- Le DAE est capable de débruiter les data, même sans avoir été entraîné sur des data propres!
- Ceci n'a rien de magique, et nous allons le montrer dans les slides suivants.
- L'hypothèse d'indépendance des deux bruits est cruciale.

Remarques

- Bien que le DAE soit entraîné sur des data bruitées uniquement, au fur et à mesure de son entraînement, il apprend à débruiter ces data de mieux en mieux.
- Le DAE est capable de débruiter les data, même sans avoir été entraîné sur des data propres!
- Ceci n'a rien de magique, et nous allons le montrer dans les slides suivants.
- L'hypothèse d'indépendance des deux bruits est cruciale.

Noise 2 Noise: preuve

- Soient $\bar{\mathbf{X}}$ et $\tilde{\mathbf{X}}$ des data bruitées provenant de data $\mathbf{X}$ par adjonction de bruits *indépendants*, *additifs* et *non-biaisés (i.e. moyennes nulles)* $\boldsymbol{\epsilon}$ et $\boldsymbol{\delta}$, e.g.:

$$\begin{aligned} \bar{\mathbf{X}} &= \mathbf{X} + \boldsymbol{\epsilon} \quad \text{où} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}) \\ \tilde{\mathbf{X}} &= \mathbf{X} + \boldsymbol{\delta} \quad \text{où} \quad \boldsymbol{\delta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}') \end{aligned}$$

- Alors la minimisation de la mean squared error (MSE)

$$\mathcal{L}(\bar{\mathbf{X}}, \tilde{\mathbf{X}}; \boldsymbol{\Theta}) = \frac{1}{N} \sum_{i=1}^{N} \left\| \mathcal{N}\left(\tilde{\mathbf{x}}_{\mathbf{i}}; \boldsymbol{\Theta}\right) - \bar{\mathbf{x}}_{\mathbf{i}} \right\|^2$$

induit un réseau de neurones $\mathcal{N}$ capable de débruiter les data.

- Comment est-ce possible?

Noise 2 Noise: preuve

- Soient $\bar{\mathbf{X}}$ et $\tilde{\mathbf{X}}$ des data bruitées provenant de data $\mathbf{X}$ par adjonction de bruits *indépendants*, *additifs* et *non-biaisés (i.e. moyennes nulles)* $\boldsymbol{\epsilon}$ et $\boldsymbol{\delta}$, e.g.:

$$\begin{aligned} \bar{\mathbf{X}} &= \mathbf{X} + \boldsymbol{\epsilon} \quad \text{où} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}) \\ \tilde{\mathbf{X}} &= \mathbf{X} + \boldsymbol{\delta} \quad \text{où} \quad \boldsymbol{\delta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}') \end{aligned}$$

- Alors la minimisation de la mean squared error (MSE)

$$\mathcal{L}(\bar{\mathbf{X}}, \tilde{\mathbf{X}}; \boldsymbol{\Theta}) = \frac{1}{N} \sum_{i=1}^{N} \left\| \mathcal{N}\left(\tilde{\mathbf{x}}_{\mathbf{i}}; \boldsymbol{\Theta}\right) - \bar{\mathbf{x}}_{\mathbf{i}} \right\|^2$$

induit un réseau de neurones $\mathcal{N}$ capable de débruiter les data.

- Comment est-ce possible?

Noise 2 Noise: preuve

- Soient $\bar{\mathbf{X}}$ et $\tilde{\mathbf{X}}$ des data bruitées provenant de data $\mathbf{X}$ par adjonction de bruits *indépendants*, *additifs* et *non-biaisés (i.e. moyennes nulles)* $\boldsymbol{\epsilon}$ et $\boldsymbol{\delta}$, e.g.:

$$\begin{aligned} \bar{\mathbf{X}} &= \mathbf{X} + \boldsymbol{\epsilon} \quad \text{où} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}) \\ \tilde{\mathbf{X}} &= \mathbf{X} + \boldsymbol{\delta} \quad \text{où} \quad \boldsymbol{\delta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}') \end{aligned}$$

- Alors la minimisation de la mean squared error (MSE)

$$\mathcal{L}(\bar{\mathbf{X}}, \tilde{\mathbf{X}}; \boldsymbol{\Theta}) = \frac{1}{N} \sum_{i=1}^{N} \left\| \mathcal{N}\left(\tilde{\mathbf{x}}_{\mathbf{i}}; \boldsymbol{\Theta}\right) - \bar{\mathbf{x}}_{\mathbf{i}} \right\|^2$$

  induit un réseau de neurones $\mathcal{N}$ capable de débruiter les data.

- Comment est-ce possible?

Noise 2 Noise: preuve

- En fait, $\mathcal{L}(\bar{\mathbf{X}}, \tilde{\mathbf{X}}; \boldsymbol{\Theta})$ est un estimateur de $\mathbb{E}\left[\|\mathcal{N}(\tilde{\mathbf{X}}; \boldsymbol{\Theta}) - \bar{\mathbf{X}}\|^2\right]$.

  De plus, on a

$$\begin{aligned}
&\mathbb{E}\left[\|\mathcal{N}(\tilde{\mathbf{X}}; \boldsymbol{\Theta}) - \bar{\mathbf{X}}\|^2\right] = \\
&\mathbb{E}\left[\|\mathcal{N}(\mathbf{X} + \delta; \boldsymbol{\Theta}) - (\mathbf{X} + \epsilon)\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \delta; \boldsymbol{\Theta}) - \mathbf{X}) - \epsilon)\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \delta; \boldsymbol{\Theta}) - \mathbf{X})\|^2\right] - 2\mathbb{E}\left[\epsilon^T(\mathcal{N}(\mathbf{X} + \delta; \boldsymbol{\Theta}) - \mathbf{X})\right] + \mathbb{E}\left[\|\epsilon\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \delta; \boldsymbol{\Theta}) - \mathbf{X})\|^2\right] - 2\underbrace{\mathbb{E}\left[\epsilon\right]^T}_{=0}\mathbb{E}\left[\mathcal{N}(\mathbf{X} + \delta; \boldsymbol{\Theta}) - \mathbf{X}\right] + \mathbb{E}\left[\|\epsilon\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \delta; \boldsymbol{\Theta}) - \mathbf{X})\|^2\right] + \mathbb{E}\left[\|\epsilon\|^2\right]
\end{aligned}$$

  et donc

$$\begin{aligned}
\hat{\boldsymbol{\Theta}} &= \arg\min_{\boldsymbol{\Theta}} \mathbb{E}\left[\|\mathcal{N}(\tilde{\mathbf{X}}; \boldsymbol{\Theta}) - \bar{\mathbf{X}}\|^2\right] \\
&= \arg\min_{\boldsymbol{\Theta}} \mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \delta; \boldsymbol{\Theta}) - \mathbf{X})\|^2\right] + \mathbb{E}\left[\|\epsilon\|^2\right] \\
&= \arg\min_{\boldsymbol{\Theta}} \mathbb{E}\left[\|(\mathcal{N}(\tilde{\mathbf{X}}; \boldsymbol{\Theta}) - \mathbf{X})\|^2\right]
\end{aligned}$$

Noise 2 Noise: preuve

- En fait, $\mathcal{L}(\bar{\mathbf{X}}, \tilde{\mathbf{X}}; \boldsymbol{\Theta})$ est un estimateur de $\mathbb{E}\left[\|\mathcal{N}(\tilde{\mathbf{X}}; \boldsymbol{\Theta}) - \bar{\mathbf{X}}\|^2\right]$.

  De plus, on a

$$
\begin{aligned}
&\mathbb{E}\left[\|\mathcal{N}(\tilde{\mathbf{X}}; \boldsymbol{\Theta}) - \bar{\mathbf{X}}\|^2\right] = \\
&\mathbb{E}\left[\|\mathcal{N}(\mathbf{X} + \delta; \boldsymbol{\Theta}) - (\mathbf{X} + \epsilon)\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \delta; \boldsymbol{\Theta}) - \mathbf{X}) - \epsilon)\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \delta; \boldsymbol{\Theta}) - \mathbf{X})\|^2\right] - 2\mathbb{E}\left[\epsilon^T(\mathcal{N}(\mathbf{X} + \delta; \boldsymbol{\Theta}) - \mathbf{X})\right] + \mathbb{E}\left[\|\epsilon\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \delta; \boldsymbol{\Theta}) - \mathbf{X})\|^2\right] - 2\underbrace{\mathbb{E}\left[\epsilon\right]^T}_{=0}\mathbb{E}\left[\mathcal{N}(\mathbf{X} + \delta; \boldsymbol{\Theta}) - \mathbf{X}\right] + \mathbb{E}\left[\|\epsilon\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \delta; \boldsymbol{\Theta}) - \mathbf{X})\|^2\right] + \mathbb{E}\left[\|\epsilon\|^2\right]
\end{aligned}
$$

  et donc

$$
\begin{aligned}
\hat{\boldsymbol{\Theta}} &= \arg\min_{\boldsymbol{\Theta}} \mathbb{E}\left[\|\mathcal{N}(\tilde{\mathbf{X}}; \boldsymbol{\Theta}) - \bar{\mathbf{X}}\|^2\right] \\
&= \arg\min_{\boldsymbol{\Theta}} \mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \delta; \boldsymbol{\Theta}) - \mathbf{X})\|^2\right] + \mathbb{E}\left[\|\epsilon\|^2\right] \\
&= \arg\min_{\boldsymbol{\Theta}} \mathbb{E}\left[\|(\mathcal{N}(\tilde{\mathbf{X}}; \boldsymbol{\Theta}) - \mathbf{X})\|^2\right]
\end{aligned}
$$

Noise 2 Noise: preuve

- En fait, $\mathcal{L}(\bar{\mathbf{X}}, \tilde{\mathbf{X}}; \mathbf{\Theta})$ est un estimateur de $\mathbb{E}\left[\|\mathcal{N}(\tilde{\mathbf{X}}; \mathbf{\Theta}) - \bar{\mathbf{X}}\|^2\right]$.

  De plus, on a

$$
\begin{aligned}
&\mathbb{E}\left[\|\mathcal{N}(\tilde{\mathbf{X}}; \mathbf{\Theta}) - \bar{\mathbf{X}}\|^2\right] = \\
&\mathbb{E}\left[\|\mathcal{N}(\mathbf{X} + \delta; \mathbf{\Theta}) - (\mathbf{X} + \epsilon)\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \delta; \mathbf{\Theta}) - \mathbf{X}) - \epsilon)\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \delta; \mathbf{\Theta}) - \mathbf{X})\|^2\right] - 2\mathbb{E}\left[\epsilon^T(\mathcal{N}(\mathbf{X} + \delta; \mathbf{\Theta}) - \mathbf{X})\right] + \mathbb{E}\left[\|\epsilon\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \delta; \mathbf{\Theta}) - \mathbf{X})\|^2\right] - 2\underbrace{\mathbb{E}\left[\epsilon\right]^T}_{=0}\mathbb{E}\left[\mathcal{N}(\mathbf{X} + \delta; \mathbf{\Theta}) - \mathbf{X}\right] + \mathbb{E}\left[\|\epsilon\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \delta; \mathbf{\Theta}) - \mathbf{X})\|^2\right] + \mathbb{E}\left[\|\epsilon\|^2\right]
\end{aligned}
$$

  et donc

$$
\begin{aligned}
\hat{\mathbf{\Theta}} &= \arg\min_{\mathbf{\Theta}} \mathbb{E}\left[\|\mathcal{N}(\tilde{\mathbf{X}}; \mathbf{\Theta}) - \bar{\mathbf{X}}\|^2\right] \\
&= \arg\min_{\mathbf{\Theta}} \mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \delta; \mathbf{\Theta}) - \mathbf{X})\|^2\right] + \mathbb{E}\left[\|\epsilon\|^2\right] \\
&= \arg\min_{\mathbf{\Theta}} \mathbb{E}\left[\|(\mathcal{N}(\tilde{\mathbf{X}}; \mathbf{\Theta}) - \mathbf{X})\|^2\right]
\end{aligned}
$$

Noise 2 Noise: preuve

- En fait, $\mathcal{L}(\bar{\mathbf{X}}, \tilde{\mathbf{X}}; \mathbf{\Theta})$ est un estimateur de $\mathbb{E}\left[\|\mathcal{N}(\tilde{\mathbf{X}}; \mathbf{\Theta}) - \bar{\mathbf{X}}\|^2\right]$.

De plus, on a

$$\mathbb{E}\left[\|\mathcal{N}(\tilde{\mathbf{X}}; \mathbf{\Theta}) - \bar{\mathbf{X}}\|^2\right] =$$
$$\mathbb{E}\left[\|\mathcal{N}(\mathbf{X}+\boldsymbol{\delta}; \mathbf{\Theta}) - (\mathbf{X}+\boldsymbol{\epsilon})\|^2\right] =$$
$$\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X}+\boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X}) - \boldsymbol{\epsilon})\|^2\right] =$$
$$\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X}+\boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X})\|^2\right] - 2\mathbb{E}\left[\boldsymbol{\epsilon}^T(\mathcal{N}(\mathbf{X}+\boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X})\right] + \mathbb{E}\left[\|\boldsymbol{\epsilon}\|^2\right] =$$
$$\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X}+\boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X})\|^2\right] - 2\underbrace{\mathbb{E}[\boldsymbol{\epsilon}]^T}_{=0}\mathbb{E}\left[\mathcal{N}(\mathbf{X}+\boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X}\right] + \mathbb{E}\left[\|\boldsymbol{\epsilon}\|^2\right] =$$
$$\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X}+\boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X})\|^2\right] + \mathbb{E}\left[\|\boldsymbol{\epsilon}\|^2\right]$$

et donc

$$\begin{aligned}
\hat{\mathbf{\Theta}} &= \arg\min_{\mathbf{\Theta}} \mathbb{E}\left[\|\mathcal{N}(\tilde{\mathbf{X}}; \mathbf{\Theta}) - \bar{\mathbf{X}}\|^2\right] \\
&= \arg\min_{\mathbf{\Theta}} \mathbb{E}\left[\|(\mathcal{N}(\mathbf{X}+\boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X})\|^2\right] + \mathbb{E}\left[\|\boldsymbol{\epsilon}\|^2\right] \\
&= \arg\min_{\mathbf{\Theta}} \mathbb{E}\left[\|(\mathcal{N}(\tilde{\mathbf{X}}; \mathbf{\Theta}) - \mathbf{X})\|^2\right]
\end{aligned}$$

Noise 2 Noise: preuve

- En fait, $\mathcal{L}(\bar{\mathbf{X}}, \tilde{\mathbf{X}}; \boldsymbol{\Theta})$ est un estimateur de $\mathbb{E}\left[\|\mathcal{N}(\tilde{\mathbf{X}}; \boldsymbol{\Theta}) - \bar{\mathbf{X}}\|^2\right]$.

De plus, on a

$$
\begin{aligned}
&\mathbb{E}\left[\|\mathcal{N}(\tilde{\mathbf{X}}; \boldsymbol{\Theta}) - \bar{\mathbf{X}}\|^2\right] = \\
&\mathbb{E}\left[\|\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \boldsymbol{\Theta}) - (\mathbf{X} + \boldsymbol{\epsilon})\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \boldsymbol{\Theta}) - \mathbf{X}) - \boldsymbol{\epsilon})\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \boldsymbol{\Theta}) - \mathbf{X})\|^2\right] - 2\mathbb{E}\left[\boldsymbol{\epsilon}^T(\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \boldsymbol{\Theta}) - \mathbf{X})\right] + \mathbb{E}\left[\|\boldsymbol{\epsilon}\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \boldsymbol{\Theta}) - \mathbf{X})\|^2\right] - 2\underbrace{\mathbb{E}\left[\boldsymbol{\epsilon}\right]^T}_{=0}\mathbb{E}\left[\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \boldsymbol{\Theta}) - \mathbf{X}\right] + \mathbb{E}\left[\|\boldsymbol{\epsilon}\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \boldsymbol{\Theta}) - \mathbf{X})\|^2\right] + \mathbb{E}\left[\|\boldsymbol{\epsilon}\|^2\right]
\end{aligned}
$$

et donc

$$
\begin{aligned}
\hat{\boldsymbol{\Theta}} &= \arg\min_{\boldsymbol{\Theta}} \mathbb{E}\left[\|\mathcal{N}(\tilde{\mathbf{X}}; \boldsymbol{\Theta}) - \bar{\mathbf{X}}\|^2\right] \\
&= \arg\min_{\boldsymbol{\Theta}} \mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \boldsymbol{\Theta}) - \mathbf{X})\|^2\right] + \mathbb{E}\left[\|\boldsymbol{\epsilon}\|^2\right] \\
&= \arg\min_{\boldsymbol{\Theta}} \mathbb{E}\left[\|(\mathcal{N}(\tilde{\mathbf{X}}; \boldsymbol{\Theta}) - \mathbf{X})\|^2\right]
\end{aligned}
$$

Noise 2 Noise: preuve

- En fait, $\mathcal{L}(\bar{\mathbf{X}}, \tilde{\mathbf{X}}; \mathbf{\Theta})$ est un estimateur de $\mathbb{E}\left[\|\mathcal{N}(\tilde{\mathbf{X}}; \mathbf{\Theta}) - \bar{\mathbf{X}}\|^2\right]$.

De plus, on a

$$
\begin{aligned}
&\mathbb{E}\left[\|\mathcal{N}(\tilde{\mathbf{X}}; \mathbf{\Theta}) - \bar{\mathbf{X}}\|^2\right] = \\
&\mathbb{E}\left[\|\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \mathbf{\Theta}) - (\mathbf{X} + \boldsymbol{\epsilon})\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X}) - \boldsymbol{\epsilon})\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X})\|^2\right] - 2\mathbb{E}\left[\boldsymbol{\epsilon}^T(\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X})\right] + \mathbb{E}\left[\|\boldsymbol{\epsilon}\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X})\|^2\right] - 2\underbrace{\mathbb{E}\left[\boldsymbol{\epsilon}\right]^T}_{=0}\mathbb{E}\left[\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X}\right] + \mathbb{E}\left[\|\boldsymbol{\epsilon}\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X})\|^2\right] + \mathbb{E}\left[\|\boldsymbol{\epsilon}\|^2\right]
\end{aligned}
$$

et donc

$$
\begin{aligned}
\hat{\mathbf{\Theta}} &= \arg\min_{\mathbf{\Theta}} \mathbb{E}\left[\|\mathcal{N}(\tilde{\mathbf{X}}; \mathbf{\Theta}) - \bar{\mathbf{X}}\|^2\right] \\
&= \arg\min_{\mathbf{\Theta}} \mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X})\|^2\right] + \mathbb{E}\left[\|\boldsymbol{\epsilon}\|^2\right] \\
&= \arg\min_{\mathbf{\Theta}} \mathbb{E}\left[\|(\mathcal{N}(\tilde{\mathbf{X}}; \mathbf{\Theta}) - \mathbf{X})\|^2\right]
\end{aligned}
$$

Noise 2 Noise: preuve

- En fait, $\mathcal{L}(\bar{\mathbf{X}}, \tilde{\mathbf{X}}; \boldsymbol{\Theta})$ est un estimateur de $\mathbb{E}\left[\|\mathcal{N}(\tilde{\mathbf{X}}; \boldsymbol{\Theta}) - \bar{\mathbf{X}}\|^2\right]$.

De plus, on a

$$
\begin{aligned}
&\mathbb{E}\left[\|\mathcal{N}(\tilde{\mathbf{X}}; \boldsymbol{\Theta}) - \bar{\mathbf{X}}\|^2\right] = \\
&\mathbb{E}\left[\|\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \boldsymbol{\Theta}) - (\mathbf{X} + \boldsymbol{\epsilon})\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \boldsymbol{\Theta}) - \mathbf{X}) - \boldsymbol{\epsilon})\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \boldsymbol{\Theta}) - \mathbf{X})\|^2\right] - 2\mathbb{E}\left[\boldsymbol{\epsilon}^T(\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \boldsymbol{\Theta}) - \mathbf{X})\right] + \mathbb{E}\left[\|\boldsymbol{\epsilon}\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \boldsymbol{\Theta}) - \mathbf{X})\|^2\right] - 2\,\underbrace{\mathbb{E}\left[\boldsymbol{\epsilon}\right]^T}_{=0}\mathbb{E}\left[\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \boldsymbol{\Theta}) - \mathbf{X}\right] + \mathbb{E}\left[\|\boldsymbol{\epsilon}\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \boldsymbol{\Theta}) - \mathbf{X})\|^2\right] + \mathbb{E}\left[\|\boldsymbol{\epsilon}\|^2\right]
\end{aligned}
$$

et donc

$$
\begin{aligned}
\hat{\boldsymbol{\Theta}} &= \arg\min_{\boldsymbol{\Theta}} \mathbb{E}\left[\|\mathcal{N}(\tilde{\mathbf{X}}; \boldsymbol{\Theta}) - \bar{\mathbf{X}}\|^2\right] \\
&= \arg\min_{\boldsymbol{\Theta}} \mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \boldsymbol{\Theta}) - \mathbf{X})\|^2\right] + \mathbb{E}\left[\|\boldsymbol{\epsilon}\|^2\right] \\
&= \arg\min_{\boldsymbol{\Theta}} \mathbb{E}\left[\|(\mathcal{N}(\tilde{\mathbf{X}}; \boldsymbol{\Theta}) - \mathbf{X})\|^2\right]
\end{aligned}
$$

Noise 2 Noise: preuve

- En fait, $\mathcal{L}(\bar{\mathbf{X}}, \tilde{\mathbf{X}}; \mathbf{\Theta})$ est un estimateur de $\mathbb{E}\left[\|\mathcal{N}(\tilde{\mathbf{X}}; \mathbf{\Theta}) - \bar{\mathbf{X}}\|^2\right]$.

De plus, on a

$$
\begin{aligned}
&\mathbb{E}\left[\|\mathcal{N}(\tilde{\mathbf{X}}; \mathbf{\Theta}) - \bar{\mathbf{X}}\|^2\right] = \\
&\mathbb{E}\left[\|\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \mathbf{\Theta}) - (\mathbf{X} + \boldsymbol{\epsilon})\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X}) - \boldsymbol{\epsilon})\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X})\|^2\right] - 2\mathbb{E}\left[\boldsymbol{\epsilon}^T(\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X})\right] + \mathbb{E}\left[\|\boldsymbol{\epsilon}\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X})\|^2\right] - 2\,\underbrace{\mathbb{E}\left[\boldsymbol{\epsilon}\right]^T}_{=0}\,\mathbb{E}\left[\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X}\right] + \mathbb{E}\left[\|\boldsymbol{\epsilon}\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X})\|^2\right] + \mathbb{E}\left[\|\boldsymbol{\epsilon}\|^2\right]
\end{aligned}
$$

et donc

$$
\begin{aligned}
\hat{\mathbf{\Theta}} &= \arg\min_{\mathbf{\Theta}} \mathbb{E}\left[\|\mathcal{N}(\tilde{\mathbf{X}}; \mathbf{\Theta}) - \bar{\mathbf{X}}\|^2\right] \\
&= \arg\min_{\mathbf{\Theta}} \mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X})\|^2\right] + \mathbb{E}\left[\|\boldsymbol{\epsilon}\|^2\right] \\
&= \arg\min_{\mathbf{\Theta}} \mathbb{E}\left[\|(\mathcal{N}(\tilde{\mathbf{X}}; \mathbf{\Theta}) - \mathbf{X})\|^2\right]
\end{aligned}
$$

Noise 2 Noise: preuve

- En fait, $\mathcal{L}(\bar{\mathbf{X}}, \tilde{\mathbf{X}}; \boldsymbol{\Theta})$ est un estimateur de $\mathbb{E}\left[\|\mathcal{N}(\tilde{\mathbf{X}}; \boldsymbol{\Theta}) - \bar{\mathbf{X}}\|^2\right]$.

De plus, on a

$$\mathbb{E}\left[\|\mathcal{N}(\tilde{\mathbf{X}}; \boldsymbol{\Theta}) - \bar{\mathbf{X}}\|^2\right] =$$
$$\mathbb{E}\left[\|\mathcal{N}(\mathbf{X}+\boldsymbol{\delta}; \boldsymbol{\Theta}) - (\mathbf{X}+\boldsymbol{\epsilon})\|^2\right] =$$
$$\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X}+\boldsymbol{\delta}; \boldsymbol{\Theta}) - \mathbf{X}) - \boldsymbol{\epsilon})\|^2\right] =$$
$$\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X}+\boldsymbol{\delta}; \boldsymbol{\Theta}) - \mathbf{X})\|^2\right] - 2\mathbb{E}\left[\boldsymbol{\epsilon}^T(\mathcal{N}(\mathbf{X}+\boldsymbol{\delta}; \boldsymbol{\Theta}) - \mathbf{X})\right] + \mathbb{E}\left[\|\boldsymbol{\epsilon}\|^2\right] =$$
$$\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X}+\boldsymbol{\delta}; \boldsymbol{\Theta}) - \mathbf{X})\|^2\right] - 2\,\underbrace{\mathbb{E}\left[\boldsymbol{\epsilon}\right]^T}_{=0}\,\mathbb{E}\left[\mathcal{N}(\mathbf{X}+\boldsymbol{\delta}; \boldsymbol{\Theta}) - \mathbf{X}\right] + \mathbb{E}\left[\|\boldsymbol{\epsilon}\|^2\right] =$$
$$\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X}+\boldsymbol{\delta}; \boldsymbol{\Theta}) - \mathbf{X})\|^2\right] + \mathbb{E}\left[\|\boldsymbol{\epsilon}\|^2\right]$$

et donc

$$\begin{aligned}
\hat{\boldsymbol{\Theta}} &= \arg\min_{\boldsymbol{\Theta}} \mathbb{E}\left[\|\mathcal{N}(\tilde{\mathbf{X}}; \boldsymbol{\Theta}) - \bar{\mathbf{X}}\|^2\right] \\
&= \arg\min_{\boldsymbol{\Theta}} \mathbb{E}\left[\|(\mathcal{N}(\mathbf{X}+\boldsymbol{\delta}; \boldsymbol{\Theta}) - \mathbf{X})\|^2\right] + \mathbb{E}\left[\|\boldsymbol{\epsilon}\|^2\right] \\
&= \arg\min_{\boldsymbol{\Theta}} \mathbb{E}\left[\|(\mathcal{N}(\tilde{\mathbf{X}}; \boldsymbol{\Theta}) - \mathbf{X})\|^2\right]
\end{aligned}$$

Noise 2 Noise: preuve

- En fait, $\mathcal{L}(\bar{\mathbf{X}}, \tilde{\mathbf{X}}; \mathbf{\Theta})$ est un estimateur de $\mathbb{E}\left[\|\mathcal{N}(\tilde{\mathbf{X}}; \mathbf{\Theta}) - \bar{\mathbf{X}}\|^2\right]$.

  De plus, on a

$$
\begin{aligned}
&\mathbb{E}\left[\|\mathcal{N}(\tilde{\mathbf{X}}; \mathbf{\Theta}) - \bar{\mathbf{X}}\|^2\right] = \\
&\mathbb{E}\left[\|\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \mathbf{\Theta}) - (\mathbf{X} + \boldsymbol{\epsilon})\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X}) - \boldsymbol{\epsilon})\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X})\|^2\right] - 2\mathbb{E}\left[\boldsymbol{\epsilon}^T(\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X})\right] + \mathbb{E}\left[\|\boldsymbol{\epsilon}\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X})\|^2\right] - 2\,\underbrace{\mathbb{E}\left[\boldsymbol{\epsilon}\right]^T}_{=0}\,\mathbb{E}\left[\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X}\right] + \mathbb{E}\left[\|\boldsymbol{\epsilon}\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X})\|^2\right] + \mathbb{E}\left[\|\boldsymbol{\epsilon}\|^2\right]
\end{aligned}
$$

  et donc

$$
\begin{aligned}
\hat{\mathbf{\Theta}} &= \arg\min_{\mathbf{\Theta}} \mathbb{E}\left[\|\mathcal{N}(\tilde{\mathbf{X}}; \mathbf{\Theta}) - \bar{\mathbf{X}}\|^2\right] \\
&= \arg\min_{\mathbf{\Theta}} \mathbb{E}\left[\|(\mathcal{N}(\mathbf{X} + \boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X})\|^2\right] + \mathbb{E}\left[\|\boldsymbol{\epsilon}\|^2\right] \\
&= \arg\min_{\mathbf{\Theta}} \mathbb{E}\left[\|(\mathcal{N}(\tilde{\mathbf{X}}; \mathbf{\Theta}) - \mathbf{X})\|^2\right]
\end{aligned}
$$

Noise 2 Noise: preuve

- En fait, $\mathcal{L}(\bar{\mathbf{X}}, \tilde{\mathbf{X}}; \mathbf{\Theta})$ est un estimateur de $\mathbb{E}\left[\|\mathcal{N}(\tilde{\mathbf{X}}; \mathbf{\Theta}) - \bar{\mathbf{X}}\|^2\right]$.

  De plus, on a

$$
\begin{aligned}
&\mathbb{E}\left[\|\mathcal{N}(\tilde{\mathbf{X}}; \mathbf{\Theta}) - \bar{\mathbf{X}}\|^2\right] = \\
&\mathbb{E}\left[\|\mathcal{N}(\mathbf{X}+\boldsymbol{\delta}; \mathbf{\Theta}) - (\mathbf{X}+\boldsymbol{\epsilon})\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X}+\boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X}) - \boldsymbol{\epsilon})\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X}+\boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X})\|^2\right] - 2\mathbb{E}\left[\boldsymbol{\epsilon}^T(\mathcal{N}(\mathbf{X}+\boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X})\right] + \mathbb{E}\left[\|\boldsymbol{\epsilon}\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X}+\boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X})\|^2\right] - 2\underbrace{\mathbb{E}\left[\boldsymbol{\epsilon}\right]^T}_{=0}\mathbb{E}\left[\mathcal{N}(\mathbf{X}+\boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X}\right] + \mathbb{E}\left[\|\boldsymbol{\epsilon}\|^2\right] = \\
&\mathbb{E}\left[\|(\mathcal{N}(\mathbf{X}+\boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X})\|^2\right] + \mathbb{E}\left[\|\boldsymbol{\epsilon}\|^2\right]
\end{aligned}
$$

  et donc

$$
\begin{aligned}
\hat{\mathbf{\Theta}} &= \arg\min_{\mathbf{\Theta}} \mathbb{E}\left[\|\mathcal{N}(\tilde{\mathbf{X}}; \mathbf{\Theta}) - \bar{\mathbf{X}}\|^2\right] \\
&= \arg\min_{\mathbf{\Theta}} \mathbb{E}\left[\|(\mathcal{N}(\mathbf{X}+\boldsymbol{\delta}; \mathbf{\Theta}) - \mathbf{X})\|^2\right] + \mathbb{E}\left[\|\boldsymbol{\epsilon}\|^2\right] \\
&= \arg\min_{\mathbf{\Theta}} \mathbb{E}\left[\|(\mathcal{N}(\tilde{\mathbf{X}}; \mathbf{\Theta}) - \mathbf{X})\|^2\right]
\end{aligned}
$$

Noise 2 Noise: preuve

- Ainsi, minimiser la MSE loss $\mathcal{L}(\bar{\mathbf{X}}, \tilde{\mathbf{X}}; \boldsymbol{\Theta})$ avec des data bruitées comme targets

$$\hat{\boldsymbol{\Theta}} = \arg\min_{\boldsymbol{\Theta}} \frac{1}{N} \sum_{i=1}^{N} \left\| \mathcal{N}\left(\tilde{\mathbf{x}}_{\mathbf{i}}; \boldsymbol{\Theta}\right) - \bar{\mathbf{x}}_{\mathbf{i}} \right\|^2$$

revient à minimiser la MSE loss $\mathcal{L}(\mathbf{X}, \tilde{\mathbf{X}}; \boldsymbol{\Theta})$ (originale) avec des data propres comme targets

$$\hat{\boldsymbol{\Theta}} = \arg\min_{\boldsymbol{\Theta}} \frac{1}{N} \sum_{i=1}^{N} \left\| \mathcal{N}\left(\tilde{\mathbf{x}}_{\mathbf{i}}; \boldsymbol{\Theta}\right) - \mathbf{x}_{\mathbf{i}} \right\|^2$$

- En d'autres termes, l'entraînement du DAE sur des targets bruitées équivaut à l'entraînement du DAE sur des targets propres, ce qui engendre donc un "débruiteur".

Noise 2 Noise: preuve

- Ainsi, minimiser la MSE loss $\mathcal{L}(\bar{\mathbf{X}}, \tilde{\mathbf{X}}; \boldsymbol{\Theta})$ avec des data bruitées comme targets

$$\hat{\boldsymbol{\Theta}} = \arg\min_{\boldsymbol{\Theta}} \frac{1}{N} \sum_{i=1}^{N} \left\| \mathcal{N}\left(\tilde{\mathbf{x}}_{\mathbf{i}}; \boldsymbol{\Theta}\right) - \bar{\mathbf{x}}_{\mathbf{i}} \right\|^2$$

revient à minimiser la MSE loss $\mathcal{L}(\mathbf{X}, \tilde{\mathbf{X}}; \boldsymbol{\Theta})$ (originale) avec des data propres comme targets

$$\hat{\boldsymbol{\Theta}} = \arg\min_{\boldsymbol{\Theta}} \frac{1}{N} \sum_{i=1}^{N} \left\| \mathcal{N}\left(\tilde{\mathbf{x}}_{\mathbf{i}}; \boldsymbol{\Theta}\right) - \mathbf{x}_{\mathbf{i}} \right\|^2$$

- En d'autres termes, l'entraînement du DAE sur des targets bruitées équivaut à l'entraînement du DAE sur des targets propres, ce qui engendre donc un "débruiteur".

Bibliographie

Fleuret, F. (2022).
Deep Learning Course.