Magda's technical notes on diffusion

Last update: January 19, 2025

This is another set of my technical notes on various ML topics. I started writing the 1st set when beginning my PhD, the 2nd set when starting my PostDoc, the 3d when starting as a professor and I have discovered, that forcing myself to take time and write these is extremely useful. All of the technical notes are available in my GitHub repo `https://github.com/mgswiss15/technotes`.

This is a working document not meant to be polished. There may be typos and other editing errors. Technical errors mean that I didn't quite understand something which I unfortunately cannot rule out.

Contents

1 Loss proofs **2**

2 Comparison DDPM vs DDIM **5**

1 Loss proofs

Last updated: January 19, 2025

This follows upon the DDIM discussion from January 15, 2025.

Let's assume a dataset $(x, y_{t=1}, y_{t=2})$ with 3 observations $(1,2,20),(2,4,40),(3,6,60)$ and a prediction model $y_t = \epsilon_\theta(x)$ for $t \in 1,2$, which we fit by l_2 regression loss.

Option 1): Consider simple linear model $\epsilon_\theta(x) = \theta x$ and loss with a hyperparameter λ weighing the two parts

$$\mathcal{L} = \underbrace{\sum_{i=1}^{3}(\theta x^{(i)} - y_1^{(i)})^2}_{\mathcal{L}_{t=1}} + \lambda \underbrace{\sum_{i=1}^{3}(\theta x^{(i)} - y_2^{(i)})^2}_{\mathcal{L}_{t=2}} \tag{1.1}$$

$$= (\theta-2)^2 + (2\theta-4)^2 + (3\theta-6)^2 + \lambda(\theta-20)^2 + \lambda(2\theta-40)^2 + \lambda(3\theta-60)^2 \tag{1.2}$$

From the first order derivative rule for the critical point we have

$$\nabla_\theta \mathcal{L} = \nabla_\theta \mathcal{L}_{t=1} + \lambda \nabla_\theta \mathcal{L}_{t=1} = 0 \tag{1.3}$$

We know that $\nabla_\theta \mathcal{L}_t = \sum_{i=1}^{3}(\theta x^{(i)} - y_t^{(i)})x^{(i)}$ and hence:

$$\begin{aligned} \nabla_\theta \mathcal{L}_{t=1} &= (\theta-2) + (4\theta-8) + (9\theta-18) = 14\theta - 28 \\ \nabla_\theta \mathcal{L}_{t=2} &= (\theta-20) + (4\theta-80) + (9\theta-180) = 14\theta - 280 \\ \nabla_\theta \mathcal{L} &= \nabla_\theta \mathcal{L}_{t=1} + \lambda \nabla_\theta \mathcal{L}_{t=2} = (14+14\lambda)\theta - (28+280\lambda) \end{aligned} \tag{1.4}$$

From this, we get the minimizing θ for each of the three losses

$$\theta_1 = 28/14 = 2, \quad \theta_2 = 280/14 = 20, \quad \theta = \frac{28+280\lambda}{14+14\lambda} = \frac{28(1+10\lambda)}{14(1+\lambda)} = \frac{2(1+10\lambda)}{1+\lambda} \quad . \tag{1.5}$$

As we can see the overall θ depends on the hyperparameter λ and how much weight it gives to the two parts of the loss

$$\theta(\lambda=1) = 22/2 = 11, \quad \theta(\lambda=2) = 42/3 = 14, \quad \theta(\lambda=0.5) = 12/1.5 = 8 \tag{1.6}$$

but unless we put $\lambda = 0$ we cannot find θ that would minimize both $\mathcal{L}_{t=1}$ and $\mathcal{L}_{t=2}$.

Option 2): We can include the time t directly into the model as $\epsilon_\theta(x,t) = \theta t x$. Most things stay the same and we now get $\nabla_\theta \mathcal{L}_t = \sum_{i=1}^{3}(\theta t x^{(i)} - y_t^{(i)}) t x^{(i)}$ so that:

$$\begin{aligned}\nabla_\theta \mathcal{L}_{t=1} &= (\theta - 2) + (4\theta - 8) + (9\theta - 18) = 14\theta - 28 \\ \nabla_\theta \mathcal{L}_{t=2} &= (4\theta - 40) + (16\theta - 160) + (36\theta - 360) = 56\theta - 560 \\ \nabla_\theta \mathcal{L} &= \nabla_\theta \mathcal{L}_{t=1} + \lambda \nabla_\theta \mathcal{L}_{t=2} = (14 + 56\lambda)\theta - (28 + 560\lambda) \end{aligned} \tag{1.7}$$

and the minimizings θ's are

$$\theta_1 = 28/14 = 2, \quad \theta_2 = 560/56 = 10, \quad \theta = \frac{28 + 560\lambda}{14 + 56\lambda} = \frac{28(1 + 20\lambda)}{14(1 + 4\lambda)} = \frac{2(1 + 20\lambda)}{1 + 4\lambda} \ . \tag{1.8}$$

and the overall θ again depends on λ

$$\theta(\lambda = 1) = 42/5 = 8.4, \quad \theta(\lambda = 2) = 82/9 = 9.1, \quad \theta(\lambda = 0.5) = 22/43 = 7.3 \ . \tag{1.9}$$

Again unless we put $\lambda = 0$ we cannot find θ that would minimize both $\mathcal{L}_{t=1}$ and $\mathcal{L}_{t=2}$.

Option 3): Finally we include the time t into the model as a learned embedding $t \to e(t) = e_t, 1, 2 \to \mathbb{R}$ as $\epsilon_\theta(x,t) = \theta x e(t)$. We now have $\nabla_\theta \mathcal{L}_t = \sum_{i=1}^{3}(\theta e_t x^{(i)} - y_t^{(i)}) e_t x^{(i)}$ so that:

$$\begin{aligned}\nabla_\theta \mathcal{L}_{t=1} &= e_1(e_1\theta - 2) + e_1(e_1 4\theta - 8) + e_1(e_1 9\theta - 18) = e_1(e_1 14\theta - 28) \\ \nabla_\theta \mathcal{L}_{t=2} &= e_2(e_2\theta - 20) + e_2(e_2 4\theta - 80) + e_2(e_2 9\theta - 180) = e_2(e_2 14\theta - 280) \\ \nabla_\theta \mathcal{L} &= \nabla_\theta \mathcal{L}_{t=1} + \lambda \nabla_\theta \mathcal{L}_{t=2} = (e_1^2 + e_2^2\lambda) 14\theta - (28 e_1 + 280 e_2 \lambda) \end{aligned} \tag{1.10}$$

and the minimizings θ's are

$$\theta_1 = \frac{28}{14 e_1}, \quad \theta_2 = \frac{280}{14 e_2}, \quad \theta = \frac{28 e_1 + 280 e_2 \lambda}{14(e_1^2 + e_2^2\lambda)} = \frac{28(e_1 + 10 e_2 \lambda)}{14(e_1^2 + e_2^2\lambda)} = \frac{2(e_1 + 10 e_2\lambda)}{e_1^2 + e_2^2\lambda} \ . \tag{1.11}$$

Clearly, in this case we can put $e_2 = 10 e_1$ to obtain $\theta_1 = \theta_2 = \frac{28}{14 e_1} = 2/e_1$ and we will also get

$$\theta = \frac{2(e_1 + 10 e_2 \lambda)}{e_1^2 + e_2^2 \lambda} = \frac{2(e_1 + 100 e_1 \lambda)}{e_1^2 + 100 e_1^2 \lambda} = \frac{2(e_1 + 100 e_1 \lambda)}{e_1(e_1 + 100 e_1 \lambda)} = 2/e_1 \ . \tag{1.12}$$

Hence in this case, when we learn both θ and the embedding $e(t)$, the hyperparameter λ is not relevant and we can minimize all three functions at the same time.

Option 4): More generally for any model $\epsilon(\theta, e_t, x)$ and any loss function $\mathcal{L} = \mathcal{L}_{t=1} + \lambda\mathcal{L}_{t=2}$ we can always find a $e_2 = f(\theta, e_1)$ such that $\arg\min_\theta \mathcal{L}_{t=1} = \arg\min_\theta \mathcal{L}_{t=2} = \arg\min_\theta \mathcal{L}$ for any λ.

2 Comparison DDPM vs DDIM

Jiaming Song, Chenlin Meng, and Stefano Ermon. "Denoising Diffusion Implicit Models". In: International Conference on Learning Representations. 2021

DDPM

This is technotes

DDIM

and this goes on

References

[1] Jiaming Song, Chenlin Meng, and Stefano Ermon. "Denoising Diffusion Implicit Models". In: International Conference on Learning Representations. 2021 (cit. on p. 5).