learn conditional densities $p(X|Y)$ with NFs

- e.g. Y = digit label, X: MNIST

  $X \sim p(X) \Rightarrow$ sample any digit

  $X \sim p(X|Y=2) \Rightarrow$ sample only "2"s

- often needed in practice: we know (or measured) Y, but want to know X

  typical setup of supervised learning

  traditional networks: point estimates $\hat{X} = r(Y)$ (ideally $\hat{X} = \arg\max_X p(X|Y)$) — r: regression or classification fct

  conditional normalizing flows: distribution of X $\hat{=}$ estimate uncertainty of X

- a autoregressive function is easy to generalize for conditioning

$$Z = f(X;Y) = \begin{pmatrix} f_1(X_1;Y) \\ f_2(X_2;X_1,Y) \\ \vdots \\ f_D(X_D, X_{1:D-1}, Y) \end{pmatrix}$$

idea: do this for all couplings

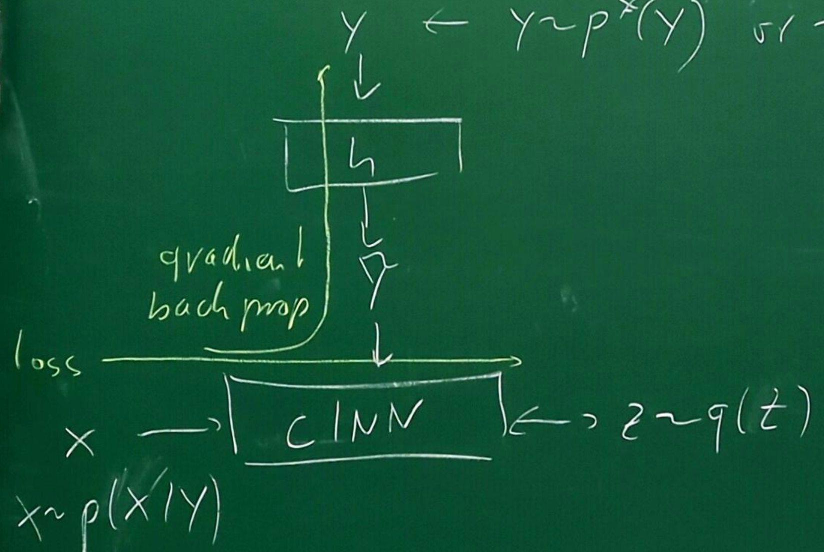
$$f_j^{(l)}\left(Z_j^{(l-1)}; Z_{1:\tilde{D}}^{(l-1)}, Y\right) = s_j^{(l)}\left(Z_{1:\tilde{D}}^{(l-1)}, Y\right) \cdot Z_j^{(l-1)} + t_j^{(l)}\left(Z_{1:\tilde{D}}^{(l-1)}, Y\right)$$

$\Rightarrow$ add Y as a new input to all nested networks $s^{(l)}, t^{(l)}$

- works because Y is known for both forward and reverse network execution

- if Y is complicated (e.g. high dimensional, image), processing Y again in each coupling layer is wasteful

$\Rightarrow$ shared preprocessing network $\tilde{Y} = h(Y)$ "feature detector" "summary network"



tricks to define $h(Y)$

- use architecture of an existing regression network
  $h(Y)$ is $r(Y)$ minus final layer(s)
- use a foundational model $\phi(Y)$

  $h(Y) = \hat{h}(\phi(Y))$

  $\hat{h}$: small

  ϕ: feature detector pretrained by the big guys on massive data

  e.g. images: CLIP, MOCO ($>10^9$ training images)

- train $f(x, h(Y))$ jointly