UNIVERSITÀ DEGLI STUDI DI FIRENZE

SCUOLA DI SCIENZE MATEMATICHE, FISICHE E NATURALI
TESI DI LAUREA TRIENNALE IN INFORMATICA

Riproduzione e analisi di un modello generativo per la creazione di immagini vettoriali

Candidato
Giuliano Gambacorta

Relatore
Prof. Paolo Frasconi

Anno Accademico 2016/2017

Outline

Outline

Principi a confronto

- Modelli discriminativi 1
  - Regressione lineare
  - Support Vector Machine (SVM)
  - ...
- Modelli generativi 2
  - Hidden Markov Model (HMM)
  - Generative Adversarial Networks (GAN)
  - Variational Autoencoder (VAE)
  - Mixture Density Network (MDN)

Principi a confronto

- Modelli discriminativi 1
  - Regressione lineare
  - Support Vector Machine (SVM)
  - ...
- Modelli generativi 2
  - Hidden Markov Model (HMM)
  - Generative Adversarial Networks (GAN)
  - Variational Autoencoder (VAE)
  - Mixture Density Network (MDN)

Principi a confronto

- Modelli discriminativi 1
  - Regressione lineare
  - Support Vector Machine (SVM)
  - ...
- Modelli generativi 2
  - Hidden Markov Model (HMM)
  - Generative Adversarial Networks (GAN)
  - Variational Autoencoder (VAE)
  - Mixture Density Network (MDN)

Principi a confronto

- Modelli discriminativi 1
  - Regressione lineare
  - Support Vector Machine (SVM)
  - ...
- Modelli generativi 2
  - Hidden Markov Model (HMM)
  - Generative Adversarial Networks (GAN)
  - Variational Autoencoder (VAE)
  - Mixture Density Network (MDN)

Principi a confronto

- Modelli discriminativi 1
  - Regressione lineare
  - Support Vector Machine (SVM)
  - ...
- Modelli generativi 2
  - Hidden Markov Model (HMM)
  - Generative Adversarial Networks (GAN)
  - Variational Autoencoder (VAE)
  - Mixture Density Network (MDN)

Formalmente:

$$f(\boldsymbol{x}) = arg \max_{y} p(\boldsymbol{y}|\boldsymbol{x}) \quad (1)$$

$$f(\boldsymbol{x}) = arg \max_{y} p(\boldsymbol{y}|\boldsymbol{x}) p(\boldsymbol{y}) \quad (2)$$

Principi a confronto

- Modelli discriminativi 1
  - Regressione lineare
  - Support Vector Machine (SVM)
  - ...
- Modelli generativi 2
  - Hidden Markov Model (HMM)
  - Generative Adversarial Networks (GAN)
  - Variational Autoencoder (VAE)
  - Mixture Density Network (MDN)

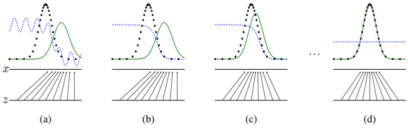
Formalmente:

$$f(\boldsymbol{x}) = arg\max_{y} p(\boldsymbol{y}|\boldsymbol{x}) \quad (1)$$

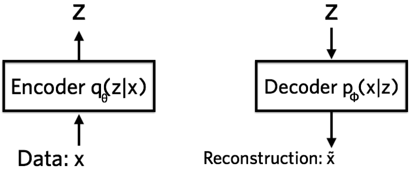
$$f(\boldsymbol{x}) = arg\max_{y} p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{y}) \quad (2)$$

Variabile latente

Esempio: GAN

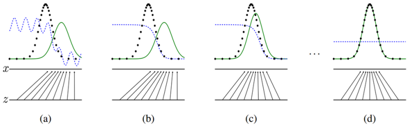


VAE



Variabile latente

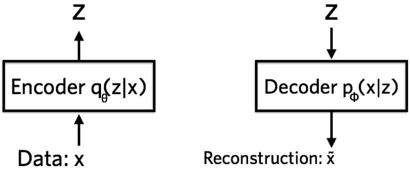
Esempio: GAN



- Reti avversarie
- **z** arbitraria
- **Loss:**

$$E_{\mathbf{x}\sim p_{data}}[\log D(\mathbf{x})] + E_{\mathbf{z}\sim p_{\mathbf{z}}}[\log(1 - D(G(\mathbf{z})))]$$

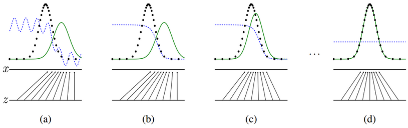
VAE



- Encoder - Decoder
- **z** condizionata
- **loss:** $-E_{\mathbf{z}\sim q}[\log(p(\mathbf{x}|\mathbf{z}))] + KL(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$

Variabile latente

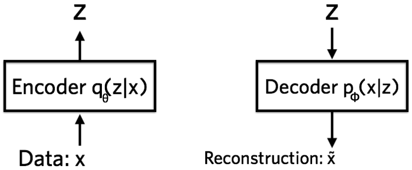
Esempio: GAN



- Reti avversarie
- **z** arbitraria
- Loss:

$$E_{\mathbf{x}\sim p_{data}}[\log D(\mathbf{x})] + E_{\mathbf{z}\sim p_{\mathbf{z}}}[\log(1 - D(G(\mathbf{z})))]$$

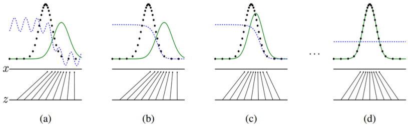
VAE



- Encoder - Decoder
- **z** condizionata
- loss: $-E_{\mathbf{z}\sim q}[\log(p(\mathbf{x}|\mathbf{z}))] + KL(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$

Variabile latente

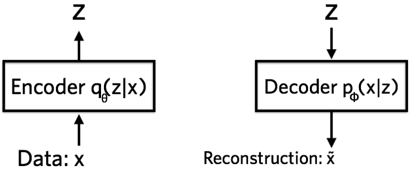
Esempio: GAN



- Reti avversarie
- **z** arbitraria
- **Loss**:

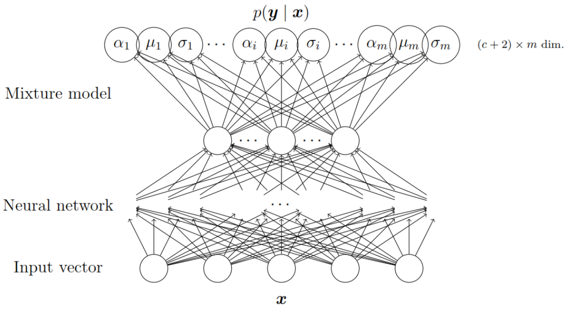
$$E_{\boldsymbol{x} \sim p_{data}}[\log D(\boldsymbol{x})] + E_{\boldsymbol{z} \sim p_{\boldsymbol{z}}}[\log(1 - D(G(\boldsymbol{z})))]$$

VAE



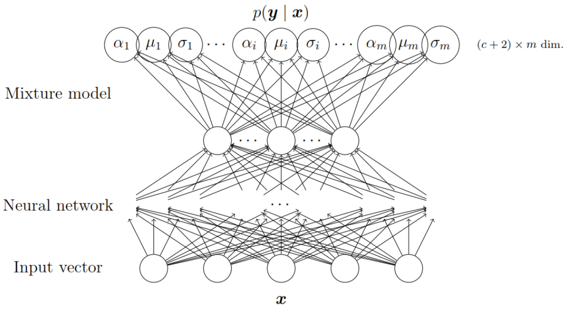
- Encoder - Decoder
- **z** condizionata
- **loss**: $-E_{\boldsymbol{z} \sim q}[\log(p(\boldsymbol{x}|\boldsymbol{z}))] + KL(q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z}))$

Mixture Density Network



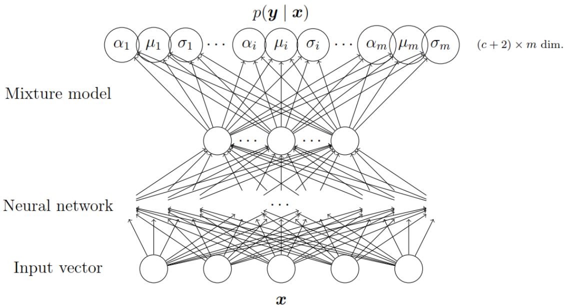
- Combina una rete neurale con una Mistura Gaussiana
- La rete fornisce i parametri della mistura
- **Loss**: $-\log(p(\boldsymbol{y}|\boldsymbol{x})) = -\log\left(\sum_{i=1}^{m} \alpha_i(\boldsymbol{x})\phi_i(\boldsymbol{y}|\boldsymbol{x})\right)$

Mixture Density Network



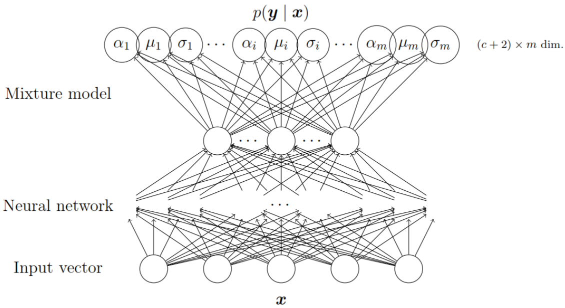
- Combina una rete neurale con una Mistura Gaussiana
- La rete fornisce i parametri della mistura
- **Loss**: $-\log(p(\boldsymbol{y}|\boldsymbol{x})) = -\log\left(\sum_{i=1}^{m} \alpha_i(\boldsymbol{x})\phi_i(\boldsymbol{y}|\boldsymbol{x})\right)$

Mixture Density Network



- Combina una rete neurale con una Mistura Gaussiana
- La rete fornisce i parametri della mistura
- **Loss**: $-\log(p(\boldsymbol{y}|\boldsymbol{x})) = -\log\left(\sum_{i=1}^{m} \alpha_i(\boldsymbol{x})\phi_i(\boldsymbol{y}|\boldsymbol{x})\right)$

Mixture Density Network



- Combina una rete neurale con una Mistura Gaussiana
- La rete fornisce i parametri della mistura
- **Loss**: $-\log(p(\boldsymbol{y}|\boldsymbol{x})) = -\log\left(\sum_{i=1}^{m} \alpha_i(\boldsymbol{x})\phi_i(\boldsymbol{y}|\boldsymbol{x})\right)$

Outline

Quick, Draw!

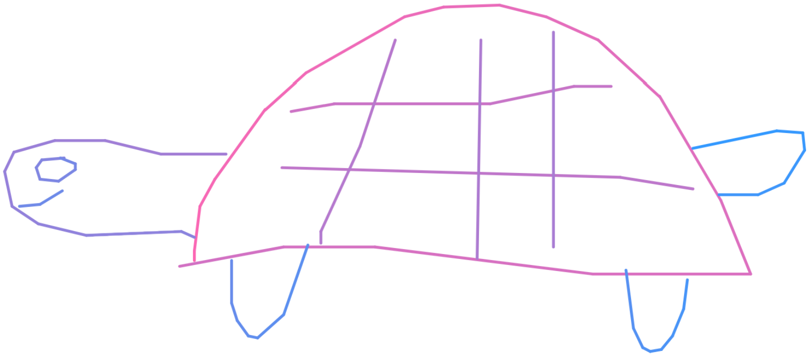
Quick, Draw!

```
[ 0,    -5,    0 ]
[ 3,    -23,   0 ]
[ 8,    -14,   0 ]
[ 27,   -36,   0 ]
[ 22,   -19,   0 ]
[ 53,   -29,   0 ]
[ 21,   -5,    0 ]
[ 30,   -1,    0 ]
[ 25,   6,     0 ]
[ 28,   12,    0 ]
[ 33,   29,    0 ]
[ 33,   54,    0 ]
[ 16,   38,    0 ]
[ -85,  0,     0 ]
[ -117, -14,   0 ]
[ -49,  0,     0 ]
[ -56,  10,    1 ]
[ 60,   -80,   0 ]
[ 23,   -4,    0 ]
[ 84,   0,     0 ]
[ 45,   -9,    0 ]
[ 20,   0,     1 ]
[ -177, 42,    0 ]
[ 182,  5,     0 ]
[ 39,   6,     1 ]
[ -160, -77,   0 ]
```

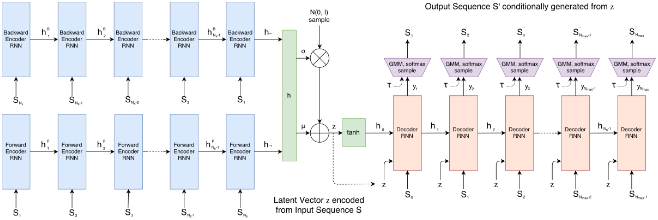
```
[ -19,  55,    0 ]
[ -21,  44,    0 ]
[ 0,    6,     1 ]
[ 86,   -105,  0 ]
[ -2,   113,   1 ]
[ 41,   -117,  0 ]
[ 0,    111,   1 ]
[ -176, -48,   0 ]
[ -35,  0,     0 ]
[ -30,  -7,    0 ]
[ -27,  0,     0 ]
[ -22,  6,     0 ]
[ -5,   10,    0 ]
[ 4,    18,    0 ]
[ 14,   9,     0 ]
[ 26,   6,     0 ]
[ 51,   -2,    0 ]
[ 7,    3,     1 ]
[ -70,  -41,   0 ]
[ -12,  1,     0 ]
[ -3,   4,     0 ]
[ 2,    6,     0 ]
[ 10,   1,     0 ]
[ 9,    -6,    0 ]
[ 0,    -3,    0 ]
[ -8,   -3,    1 ]
```

```
[ -22,  25,    0 ]
[ 11,   -1,    0 ]
[ 12,   -7,    1 ]
[ 91,   36,    0 ]
[ 0,    22,    0 ]
[ 3,    9,     0 ]
[ 6,    8,     0 ]
[ 5,    1,     0 ]
[ 14,   -12,   0 ]
[ 13,   -36,   1 ]
[ 171,  13,    0 ]
[ 4,    30,    0 ]
[ 5,    10,    0 ]
[ 4,    2,     0 ]
[ 6,    -1,    0 ]
[ 6,    -7,    0 ]
[ 6,    -14,   0 ]
[ 2,    -15,   1 ]
[ 17,   -44,   0 ]
[ 21,   0,     0 ]
[ 14,   -6,    0 ]
[ 11,   -17,   0 ]
[ -1,   -9,    0 ]
[ -14,  -1,    0 ]
[ -45,  9,     1 ]
```

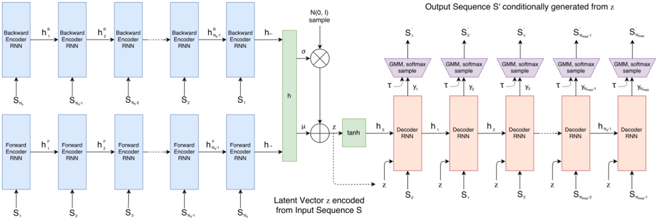
Quick, Draw!



Struttura


Output Sequence S' conditionally generated from z
N(0, I) sample
Backward Encoder RNN
Forward Encoder RNN
GMM, softmax sample
Decoder RNN
tanh
Latent Vector z encoded from Input Sequence S

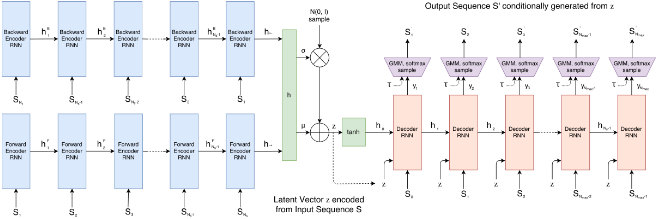
Struttura



Encoder

- Bidirectional LSTM
- Rete densamente connessa: μ
- Rete densamente connessa: $\log \sigma$

Struttura



Encoder

- Bidirectional LSTM
- Rete densamente connessa: μ
- Rete densamente connessa: $\log \sigma$

Decoder (MDN)

- tanh inizializza gli stati della LSTM
- LSTM (autoregressiva)
- Gaussian Mixture Model
- $q_k = \frac{\exp(\hat{q}_k)}{\sum_{j=1}^{3} \exp(\hat{q}_j)}, k \in 1,2,3$

Loss

VAE:

$$L = -E_{\boldsymbol{z} \sim Q}[\log(P(\boldsymbol{x}|\boldsymbol{z}))] + KL(Q(\boldsymbol{z}|\boldsymbol{x})||P(\boldsymbol{z})) \quad (3)$$

Loss

VAE:

$$L = -E_{\boldsymbol{z} \sim Q}[\log(P(\boldsymbol{x}|\boldsymbol{z}))] + KL(Q(\boldsymbol{z}|\boldsymbol{x})||P(\boldsymbol{z})) \tag{3}$$

MDN:

$$L = -\log\left(\sum_{i=1}^{m} \alpha_i(\boldsymbol{x})\phi_i(\boldsymbol{y}|\boldsymbol{x})\right) \tag{4}$$

Loss

VAE:

$$L = -E_{\boldsymbol{z}\sim Q}[\log(P(\boldsymbol{x}|\boldsymbol{z}))] + KL(Q(\boldsymbol{z}|\boldsymbol{x})||P(\boldsymbol{z})) \tag{3}$$

MDN:

$$L = -\log\left(\sum_{i=1}^{m} \alpha_i(\boldsymbol{x})\phi_i(\boldsymbol{y}|\boldsymbol{x})\right) \tag{4}$$

Combinando:

- Ricostruzione (MDN + softmax): $L_R = L_s + L_p$

$$L_s = -\frac{1}{N_{max}} \sum_{i=1}^{N_S} \log\left(\sum_{j=1}^{M} \Pi_{j,i} \mathcal{N}(\Delta x_i, \Delta y_i | \mu_{x,j,i}, \mu_{y,j,i}, \sigma_{x,j,i}, \sigma_{y,j,i}, \rho_{xy,j,i})\right) \tag{5}$$

$$L_p = -\frac{1}{N_{max}} \sum_{i=1}^{N_{max}} \sum_{k=1}^{3} p_{k,i} \log(q_{k,i}) \tag{6}$$

- Divergenza Kullback-Leibler

$$L_{KL} = -\frac{1}{2N_z}(1 + \hat{\sigma} - \mu^2 - \exp(\hat{\sigma})) \tag{7}$$

- Loss: $L_R + w_{KL} L_{KL}$

Loss

VAE:

$$L = -E_{\mathbf{z}\sim Q}[\log(P(\mathbf{x}|\mathbf{z}))] + KL(Q(\mathbf{z}|\mathbf{x})||P(\mathbf{z})) \tag{3}$$

MDN:

$$L = -\log\left(\sum_{i=1}^{m} \alpha_i(\mathbf{x})\phi_i(\mathbf{y}|\mathbf{x})\right) \tag{4}$$

Combinando:

- Ricostruzione (MDN + softmax): $L_R = L_s + L_p$

$$L_s = -\frac{1}{N_{max}} \sum_{i=1}^{N_S} \log(\sum_{j=1}^{M} \Pi_{j,i} \mathcal{N}(\Delta x_i, \Delta y_i | \mu_{x,j,i}, \mu_{y,j,i}, \sigma_{x,j,i}, \sigma_{y,j,i}, \rho_{xy,j,i})) \tag{5}$$

$$L_p = -\frac{1}{N_{max}} \sum_{i=1}^{N_{max}} \sum_{k=1}^{3} p_{k,i} \log(q_{k,i}) \tag{6}$$

- Divergenza Kullback-Leibler

$$L_{KL} = -\frac{1}{2N_z}(1 + \hat{\sigma} - \mu^2 - \exp(\hat{\sigma})) \tag{7}$$

- Loss: $L_R + w_{KL} L_{KL}$

Loss

VAE:

$$L = -E_{\boldsymbol{z}\sim Q}[\log(P(\boldsymbol{x}|\boldsymbol{z}))] + KL(Q(\boldsymbol{z}|\boldsymbol{x})||P(\boldsymbol{z})) \tag{3}$$

MDN:

$$L = -\log\left(\sum_{i=1}^{m} \alpha_i(\boldsymbol{x})\phi_i(\boldsymbol{y}|\boldsymbol{x})\right) \tag{4}$$

Combinando:

- Ricostruzione (MDN + softmax): $L_R = L_s + L_p$

$$L_s = -\frac{1}{N_{max}} \sum_{i=1}^{N_S} \log\left(\sum_{j=1}^{M} \Pi_{j,i}\mathcal{N}(\Delta x_i, \Delta y_i | \mu_{x,j,i}, \mu_{y,j,i}, \sigma_{x,j,i}, \sigma_{y,j,i}, \rho_{xy,j,i})\right) \tag{5}$$

$$L_p = -\frac{1}{N_{max}} \sum_{i=1}^{N_{max}} \sum_{k=1}^{3} p_{k,i}\log(q_{k,i}) \tag{6}$$

- Divergenza Kullback-Leibler

$$L_{KL} = -\frac{1}{2N_z}(1 + \hat{\sigma} - \mu^2 - \exp(\hat{\sigma})) \tag{7}$$

- Loss: $L_R + w_{KL}L_{KL}$

Loss

VAE:

$$L = -E_{\boldsymbol{z}\sim Q}[\log(P(\boldsymbol{x}|\boldsymbol{z}))] + KL(Q(\boldsymbol{z}|\boldsymbol{x})||P(\boldsymbol{z})) \tag{3}$$

MDN:

$$L = -\log\left(\sum_{i=1}^{m} \alpha_i(\boldsymbol{x})\phi_i(\boldsymbol{y}|\boldsymbol{x})\right) \tag{4}$$

Combinando:

- Ricostruzione (MDN + softmax): $L_R = L_s + L_p$

$$L_s = -\frac{1}{N_{max}} \sum_{i=1}^{N_S} \log\left(\sum_{j=1}^{M} \Pi_{j,i} \mathcal{N}(\Delta x_i, \Delta y_i | \mu_{x,j,i}, \mu_{y,j,i}, \sigma_{x,j,i}, \sigma_{y,j,i}, \rho_{xy,j,i})\right) \tag{5}$$

$$L_p = -\frac{1}{N_{max}} \sum_{i=1}^{N_{max}} \sum_{k=1}^{3} p_{k,i} \log(q_{k,i}) \tag{6}$$

- Divergenza Kullback-Leibler

$$L_{KL} = -\frac{1}{2N_z}(1 + \hat{\sigma} - \mu^2 - \exp(\hat{\sigma})) \tag{7}$$

- Loss: $L_R + w_{KL} L_{KL}$

Outline

Generazioni originali

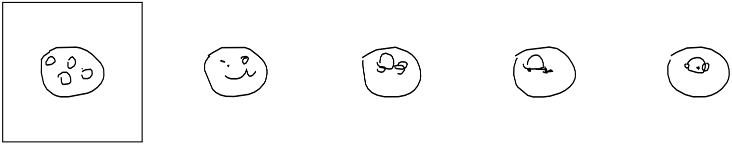
LA RETE PRODUCE IMMAGINI UNICHE E ORIGINALI

Generazioni originali

Si noti la figura umana in basso a destra ed i tentativi di ricostruirla

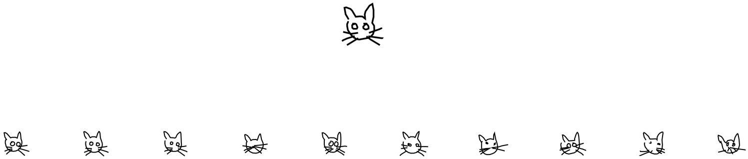
Generazioni originali

INTERPRETA SECONDO I SUOI CANONI FIGURE SCONOSCIUTE

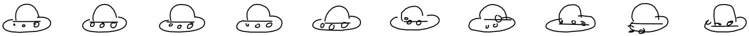


Generazioni originali

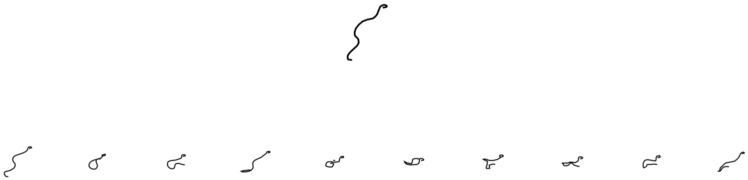
Temperature



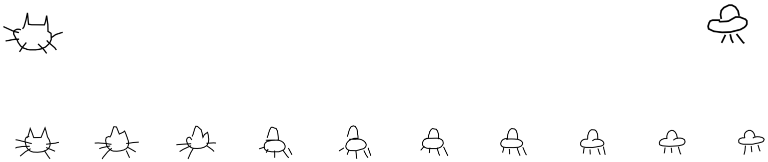
Temperature



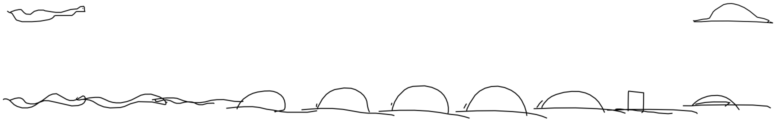
Temperature



Interpolazioni



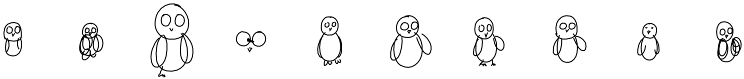
Interpolazioni



Interpolazioni

Temperature

$$\tau = 0.2$$



Temperature

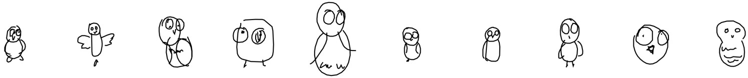
$$\tau = 0.4$$

Temperature

$$\tau = 0.6$$

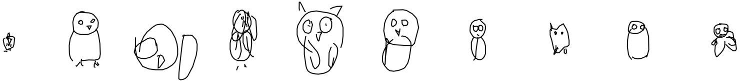
Temperature

$$\tau = 0.8$$



Temperature

$\tau = 1$



Outline

Perché valorizzare la divergenza KL?

Dataset	$w_{KL} = 1$		$w_{KL} = 0.5$		$w_{KL} = 0.25$		Decoder only
	L_R	L_{KL}	L_R	L_{KL}	L_R	L_{KL}	L_R
cat	-0.98	0.29	-1.33	0.70	-1.46	1.01	-0.57
pig	-1.14	0.22	-1.37	0.49	-1.52	0.80	-0.82
cat, pig	-1.02	0.22	-1.24	0.49	-1.50	0.98	-0.75
crab, face, pig, rabbit	-0.91	0.22	-1.04	0.40	-1.47	1.17	-0.67
face	-1.13	0.27	-1.55	0.71	-1.90	1.44	-0.73
firetruck	-1.24	0.22	-1.26	0.24	-1.78	1.10	-0.90
garden	-0.79	0.20	-0.81	0.25	-0.99	0.54	-0.62
owl	-0.93	0.20	-1.03	0.34	-1.29	0.77	-0.66
mosquito	-0.67	0.30	-1.02	0.66	-1.41	1.54	-0.34
yoga	-0.80	0.24	-1.07	0.55	-1.51	1.33	-0.48

Perché valorizzare la divergenza KL?

Outline

Progetti correlati

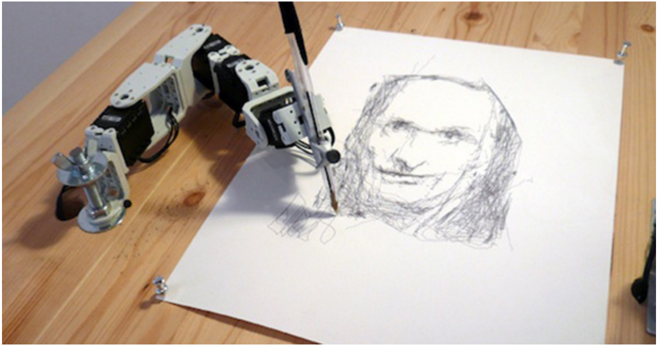
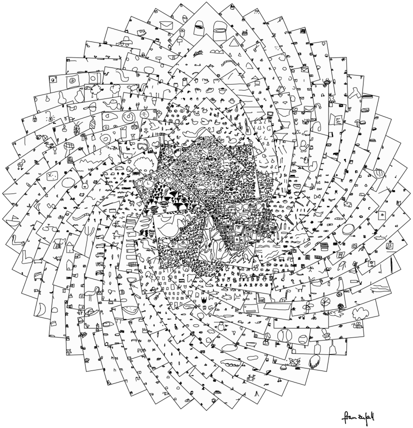


Figura: Paul the Robot

Progetti correlati



Sviluppi futuri

- Didattica
  - Disegno
  - Disturbi dell'apprendimento
- Arte
  - Assistenza del processo creativo
- Tecnica
  - Assistenza alla progettazione
- Scienza
  - Rappresentare il mondo reale

GRAZIE

PER L'ATTENZIONE!