AGH University of Science and Technology

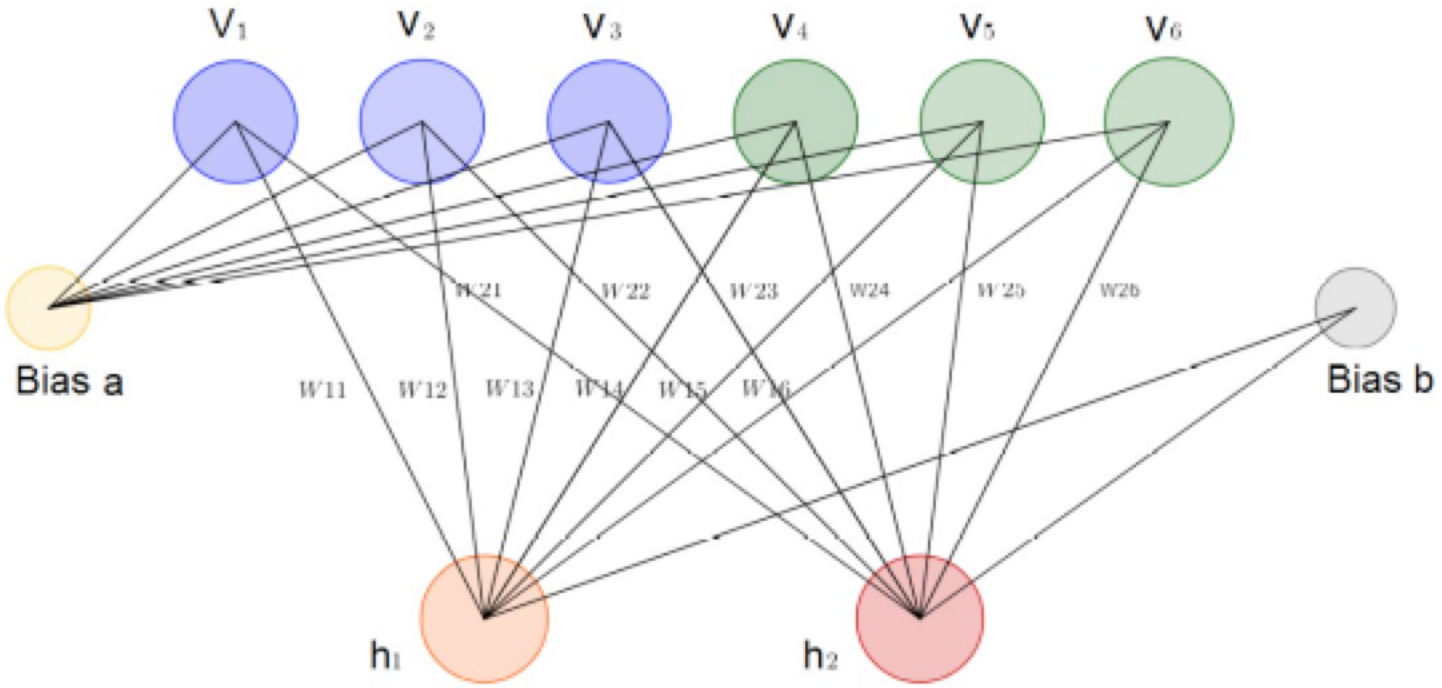
Restricted Boltzmann Machine & Deep Belief Network

Dariusz Kucharski
Katedra Automatyki i Robotyki

Kraków, 07.11.2019

Ograniczona maszyna Boltzmanna

- **Ogólna architektura sieci**



Architektura ograniczonej maszyny Boltzmanna [2]

Ograniczona maszyna Boltzmanna

- **Model bazowany na energii**

$$E(\mathbf{v}, \mathbf{h}) = -\sum_i a_i v_i - \sum_j b_j h_j - \sum_{i,j} v_i h_j w_{ij}$$

Energia ograniczone maszyny boltzmana [2]

v – wektor wejściowy,

h – odpowiedź warstwy ukrytej

w – wagi sieci

a, b - bias

Ograniczona maszyna Boltzmanna

- **Dążymy do minimalizacji energii modelu**
- **Rozkład prawdopodobieństwa wejścia i rekonstrukcji określony jest jako**

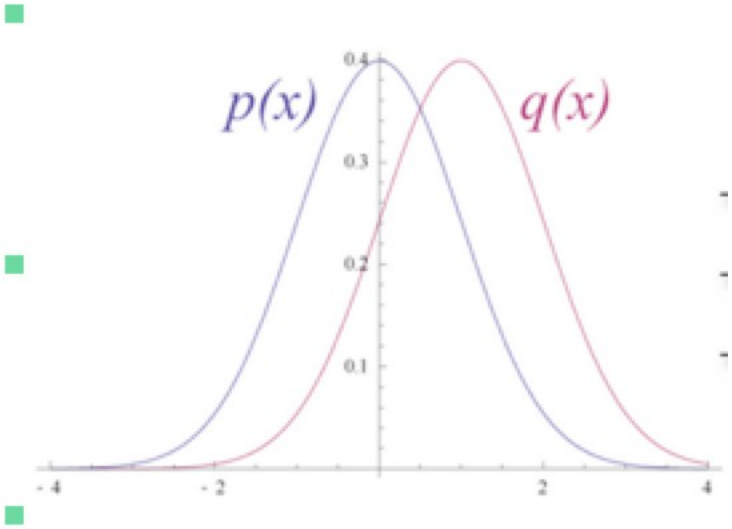
$$p(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} e^{-E(\mathbf{v},\mathbf{h})}$$

$$Z = \sum_{\mathbf{v},\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}$$

Prawdopodobieństwo łączone [2]

Ograniczona maszyna Boltzmanna

- Dążymy do minimalizacji energii modelu



Prawdopodobieństwo łączone [2]

Ograniczona maszyna Boltzmanna

- **Można zapisać jako sumę prawdopodobieństw warunkowych**

$$p(\mathbf{h}|\mathbf{v}) = \prod_i p(h_i|\mathbf{v})$$

$$p(\mathbf{v}|\mathbf{h}) = \prod_i p(v_i|\mathbf{h})$$

Prawdopodobieństwa warunkowe dla wejścia i warstwy ukrytej [2]

Ograniczona maszyna Boltzmanna

$$p(h_j = 1|\mathbf{v}) = \frac{1}{1 + e^{(-(b_j + W_j v_i))}} = \sigma(b_j + \sum_i v_i w_{ij})$$

Rozkład prawdopodobieństwa odpowiedzi warstwy ukrytej [2]

$$p(v_i = 1|\mathbf{h}) = \frac{1}{1 + e^{(-(a_i + W_i h_j))}} = \sigma(a_i + \sum_j h_j w_{ij})$$

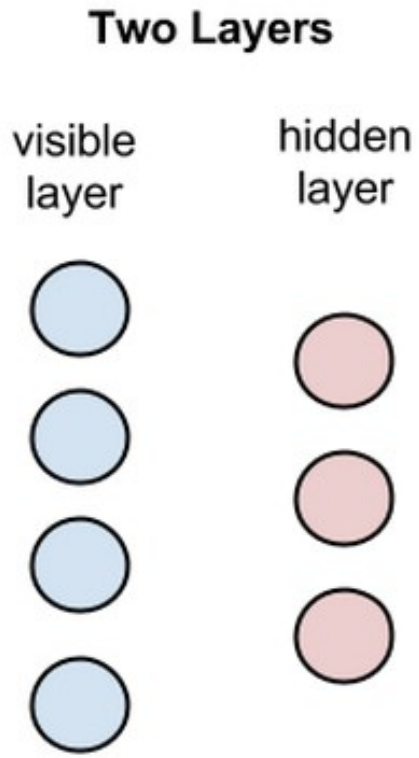
Rozkład prawdopodobieństwa rekonstrukcji [2]

Ograniczona maszyna Boltzmanna

- **Najważniejsze wnioski:**
  - **Funkcja aktywacji to sigmoid**
  - **Na wejściu jest wektor wartości binarnych**
  - **Najpierw obliczamy wartości odpowiedzi warstwy ukrytej, następnie z wykorzystaniem tych samych wag liczymy rekonstrukcję**
  - **Aktualizujemy wagi po takim przejściu tam i z powrotem (jak?)**

Ograniczona maszyna Boltzmanna

- Płytka, dwu warstwowa sieć
  - Pierwsza warstwa to warstwa widzialna (wejściowa)
  - Warstwa ukryta
  - Brak warstwy wyjściowej
- Brak połączeń między neuronami w jednej warstwie (ograniczenie)

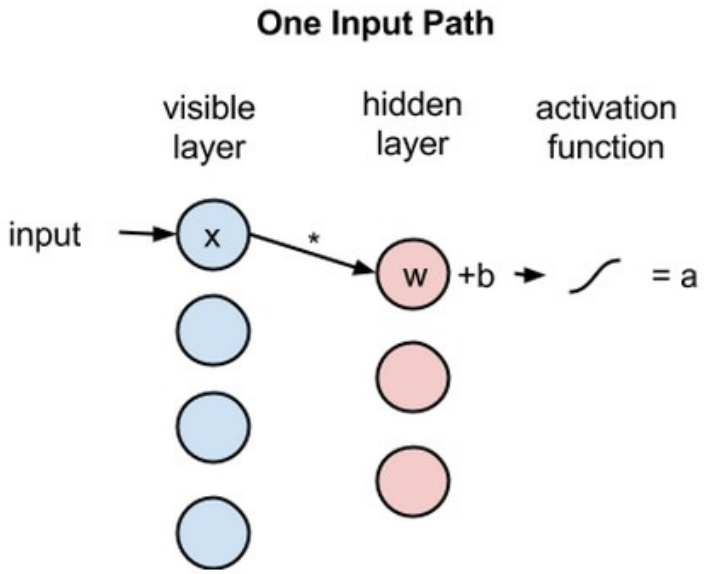


Maszyna Boltzmana [1]

Ograniczona maszyna Boltzmanna

- **Wejście jest mnożone razy wagi w**
- **Dodawany jest bias**
- **Całość poddawana jest funkcji aktywacji**
- **Wynik jest zapisywany**

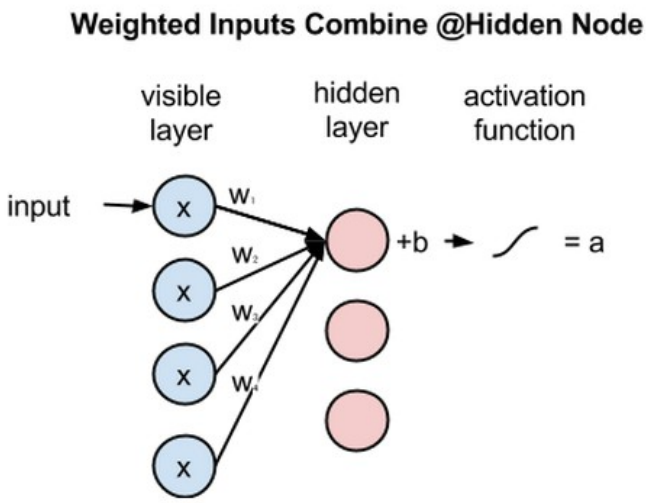
$$a = f\left(\left(weight * input\right) + bias\right)$$



Przepływ informacji [1]

Ograniczona maszyna Boltzmanna

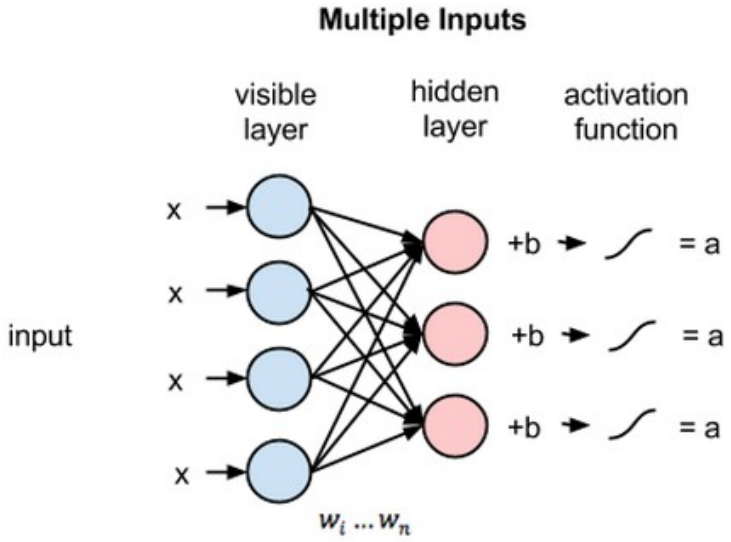
- **Sumowane jest ważone wejście z każdego elementu wektora cech**



Przepływ informacji [1]

Ograniczona maszyna Boltzmanna

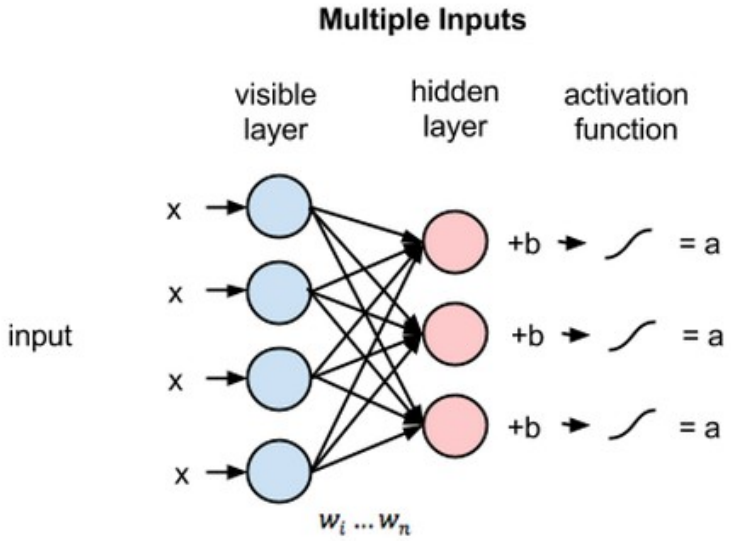
- Suma ważona dla każdego neuronu w warstwie ukrytej
- Wynikiem jest odpowiedź warstwy ukrytej
- Każe wejście jest mnożone przez wagę na każdym z ukrytych neuronów
- Czyli np. dla wejścia 4 elementowego i 3 neuronów ukrytych, macierz wag ma wymiar 4x3



Przepływ informacji [1]

Ograniczona maszyna Boltzmanna

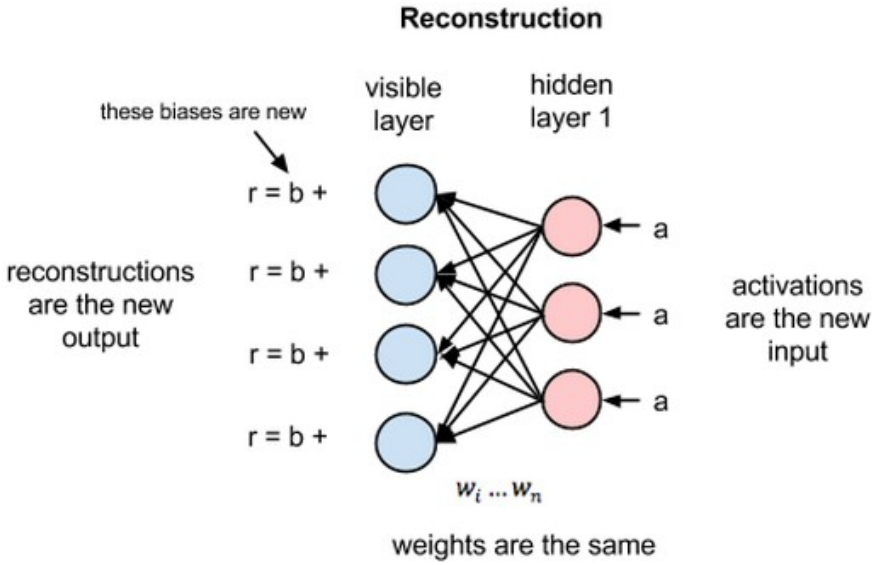
- $X_{1x4} * W_{4x3} = a_{1x3}$
- **Wynik a jest funkcją prawdopodobieństwa**
- **Warstwa ukryta przedstawia rozkład prawdopodobieństwa dla wektorów danych wejściowych**



Przepływ informacji [1]

Ograniczona maszyna Boltzmanna

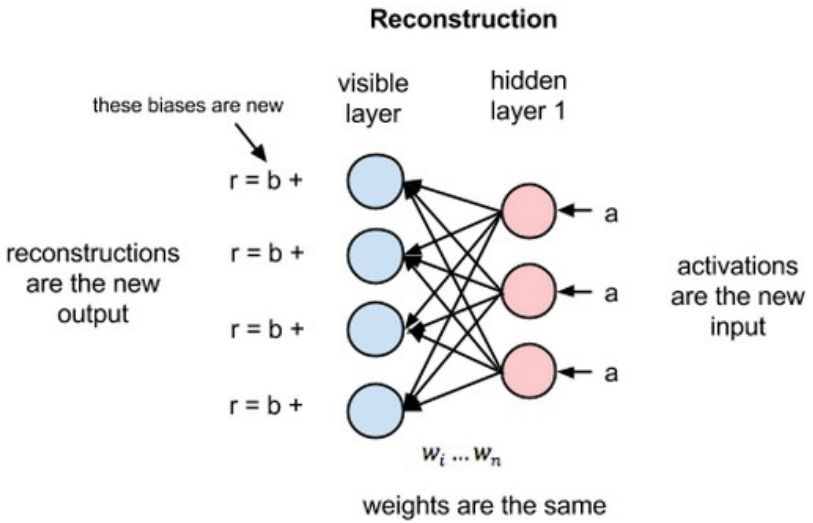
- $X_{1x4} * W_{4x3} + bx_{1x3} = a_{1x3}$
- $a_{1x3} * W^{T}_{3x4} + by_{1x4} = X'_{1x4}$



Przepływ wsteczny [1]

Ograniczona maszyna Boltzmanna

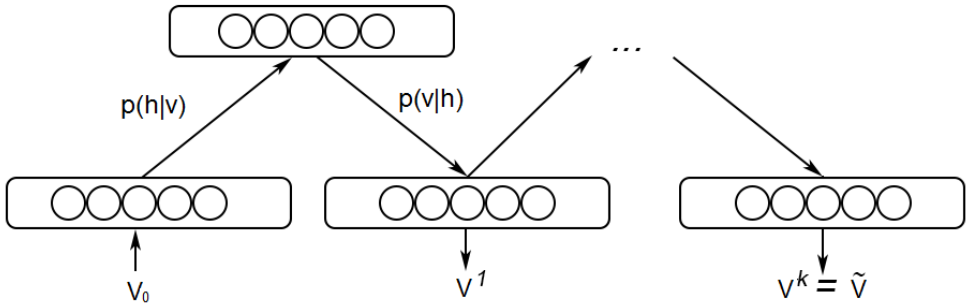
- **Wagi w są losowe, dlatego rekonstrukcja w początkowej fazie jest dość słaba**
- **Wykorzystując rekonstrukcję x’ (r), można wykonać następne przejście**
- **Po wykonaniu i iteracji (choć w praktyce jedna w zupełności wystarcza) należy policzyć błąd rekonstrukcji x’-x**



Przepływ wsteczny [1]

Ograniczona maszyna Boltzmanna

- **Wagi w są losowe, dlatego rekonstrukcja w początkowej fazie jest dość słaba**
- **Wykorzystując rekonstrukcję x' (r), można wykonać następne przejście**



Próbkowanie Gibbsa [2]

Ograniczona maszyna Boltzmanna

- **Macierz aktualizacji wag liczy się za pomocą tzw. Contrastive Divergence**
- **Wykorzystuje się przy tym iloczyn diadyczny**

$$\triangle W = \mathbf{v}_0 \otimes p(\mathbf{h}_0|\mathbf{v}_0) - \mathbf{v}_k \otimes p(\mathbf{h}_k|\mathbf{v}_k)$$

Contrastive divergence [2]

$$\mathbf{u} \otimes \mathbf{v}^T = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \otimes \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix} = \begin{bmatrix} u_1 v_1 & u_1 v_2 & u_1 v_3 \\ u_2 v_1 & u_2 v_2 & u_2 v_3 \\ u_3 v_1 & u_3 v_2 & u_3 v_3 \end{bmatrix}$$

Iloczyn diadyczny (outer product) [3]

Ograniczona maszyna Boltzmanna

- Choć można również policzyć x’-x i propagować wstecznie błąd rekonstrukcji
- Tak skonstruowana ograniczona maszyna boltzmana jest odpowiednikiem autoenkodera z jedną warstwą ukrytą z funkcją aktywacji sigmoid, związanymi wagami dekodera i enkodera na początku, gdzie na wejściu są wartości binarne

$$W_{new} = W_{old} + \triangle W$$

Aktualizacja wag [3]

Autoenkodery

Sieć neuronowa, używana do efektywnego kodowania danych wejściowych (odnalezienia reprezentacji charakteryzującej dany zbiór, często o mniejszym wymiarze niż wejście) w sposób nienadzorowany.

Możemy wyróżnić enkoder (część sieci odpowiedzialna za kodowanie), który dokonuje transformacji wejścia do pewnej ukrytej reprezentacji z (stanowiącej pewnego rodzaju kod):

$$z = f_\theta(x) = \sigma(Wx + b)$$

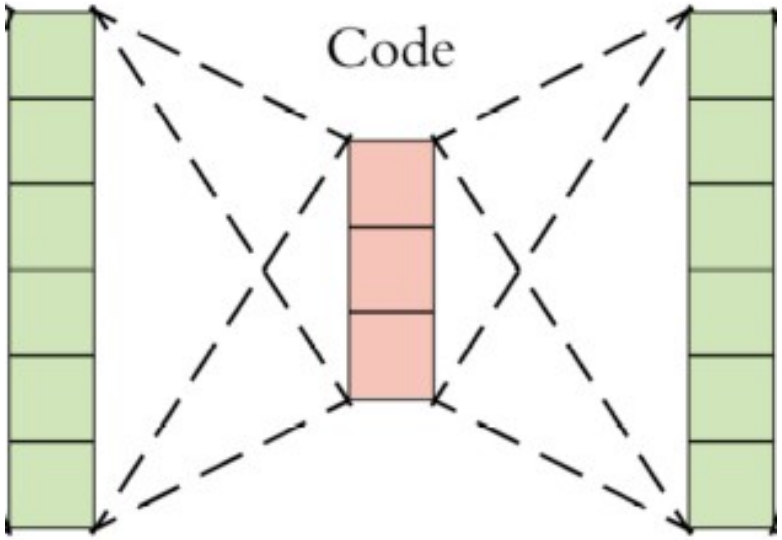
Oraz dekoder, który próbuje otworzyć te dane wejściowe na podstawie ich ukrytej reprezentacji z

$$x' = f_{\theta'}(h) = \sigma'(W'z + b')$$

Minimalizując funkcję błędu między wejściem i rekonstrukcją.

W, W', b, b' - parametry sieci; σ, σ' – funkcje aktywacji

Autoenkodery



https://towardsdatascience.com/generating-images-with-autoencoders-77fd3a8dd368

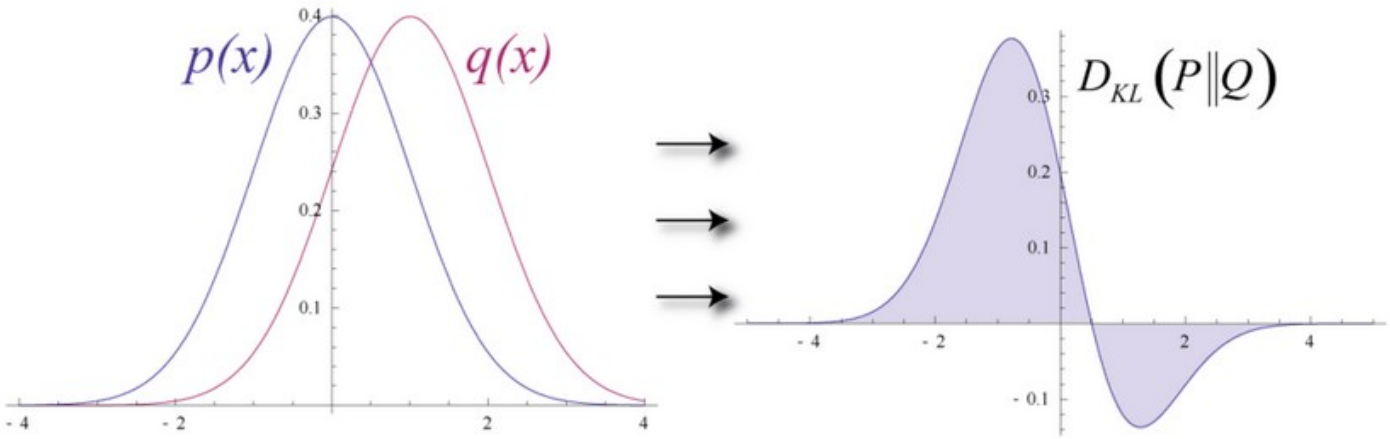
Dla W’ = W^T w początkowej fazie uczenia – połowa mniej parametrów do optymalizacji (tied weights autoencoder)

Ograniczona maszyna Boltzmanna

- Odpowiedzi warstwy ukrytej interpretujemy jako prawdopodobieństwo p(a|x; w)
- Przy obliczeniu rekonstrukcji możemy wynik rekonstrukcji interpretować jako p(x|a; w)
- Suma tych prawdopodobieństw to wspólny rozkład prawdopodobieństwa p(a, x) (joint probability)

Ograniczona maszyna Boltzmanna

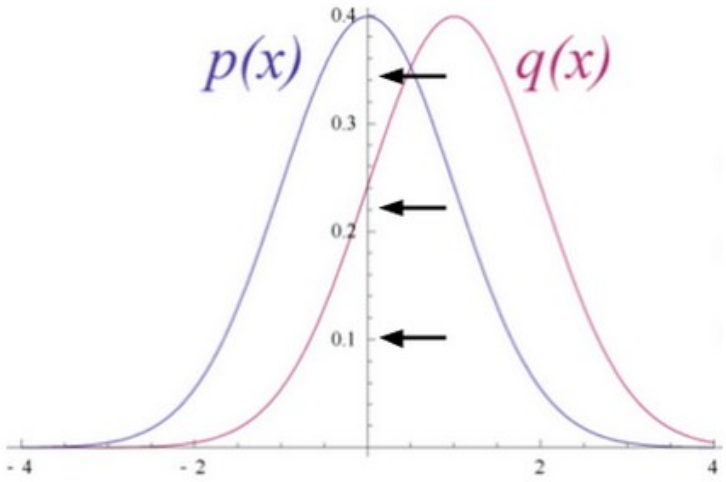
- **Jeśli klasyfikacja to odgadywanie klasy (discriminative learning), regresja to szacowanie liczby, to rekonstrukcja to szukanie rozkładu prawdopodobieństwa wejścia (generative learning)**



Różnica w rozkładzie prawdopodobieństwa sygnału oryginalnego i rekonstrukcji [1]

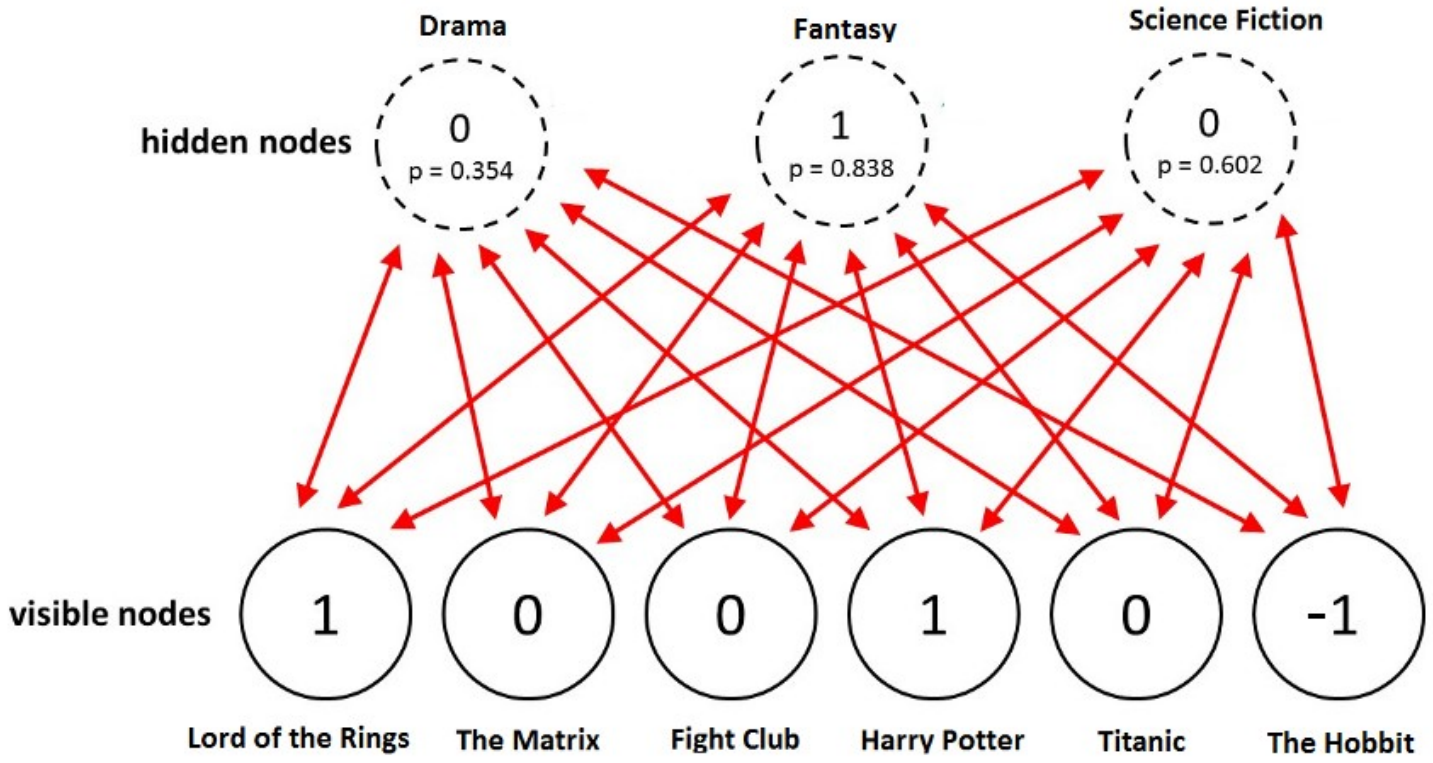
Ograniczona maszyna Boltzmanna

- **Jeśli klasyfikacja to odgadywanie klasy (discriminative learning), regresja to szacowanie liczby, to rekonstrukcja to szukanie rozkładu prawdopodobieństwa wejścia (generative learning)**



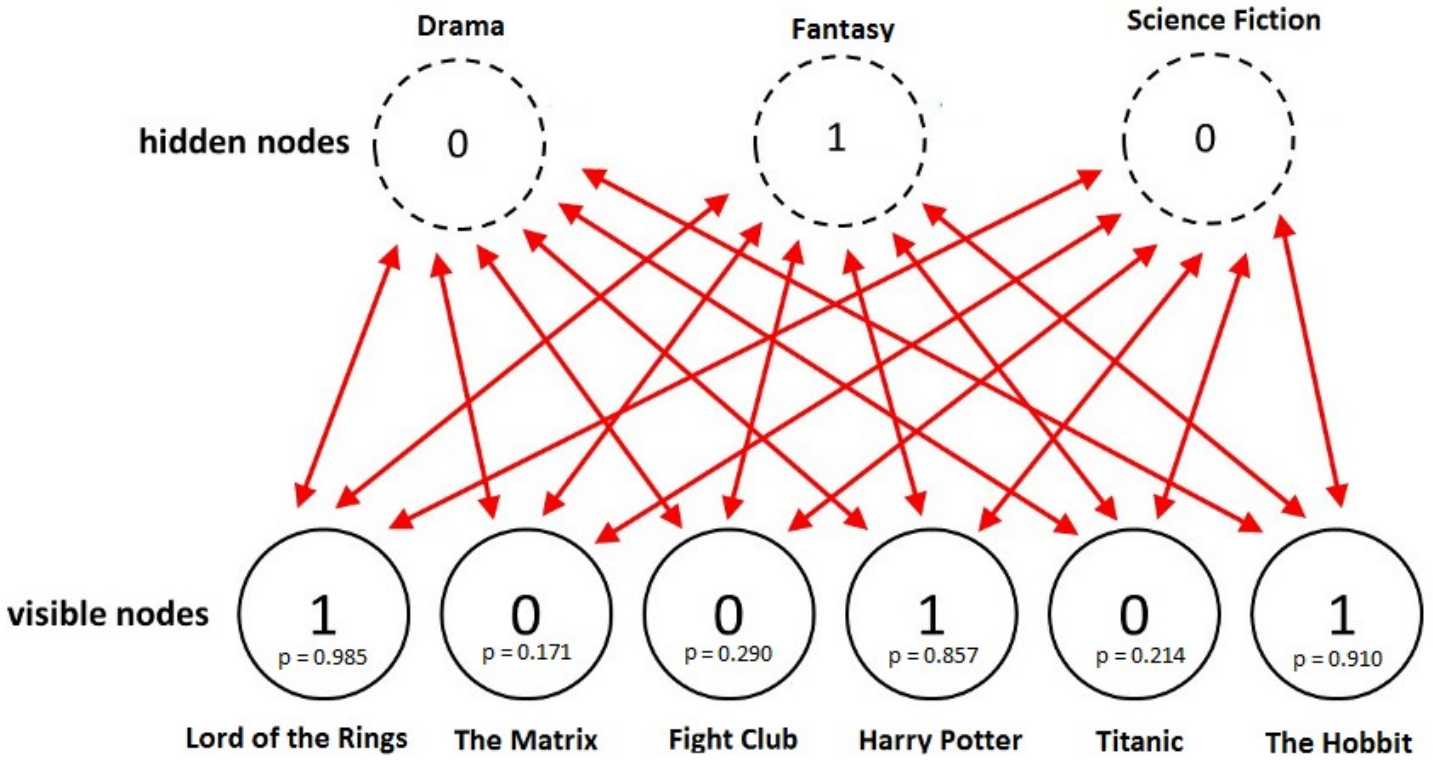
Dążymy to minimalizacji tej rekonstrukcji [1]

Ograniczona maszyna Boltzmanna



Przykład zastosowania [2]

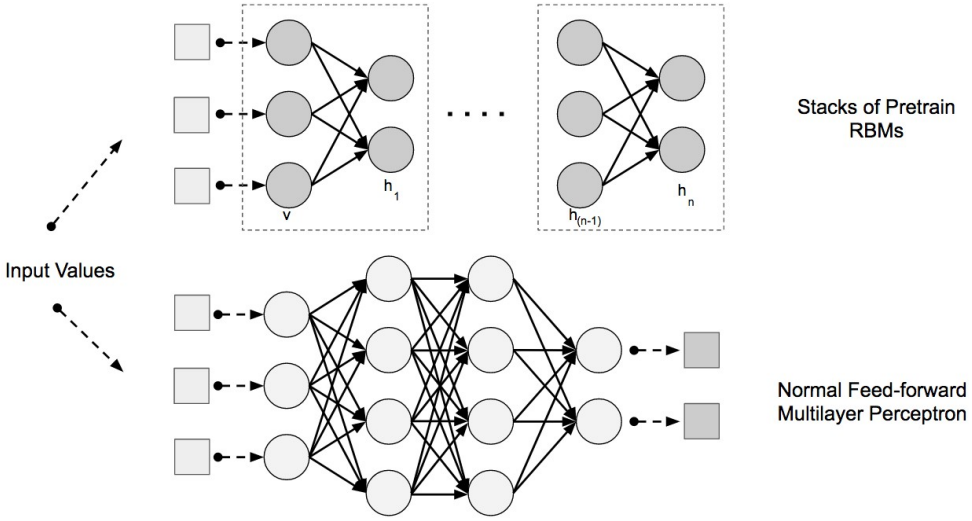
Ograniczona maszyna Boltzmanna



Przykład zastosowania [2]

Deep Belief Network

- Powstaje poprzez wstępnej uczenie Ograniczonych Maszyn Boltzmanna
- Wejście na kolejną maszynę boltzmana jest wyjściem z poprzedniej
- Po wyuczeniu odpowiedniej ilości warstw, ograniczone maszyny boltzmana łączone są w sieć głęboką (często dodaje się też klasyfikator) i następuję fine tuning – douczenie sieci, aby otrzymać jeszcze lepszą skuteczność



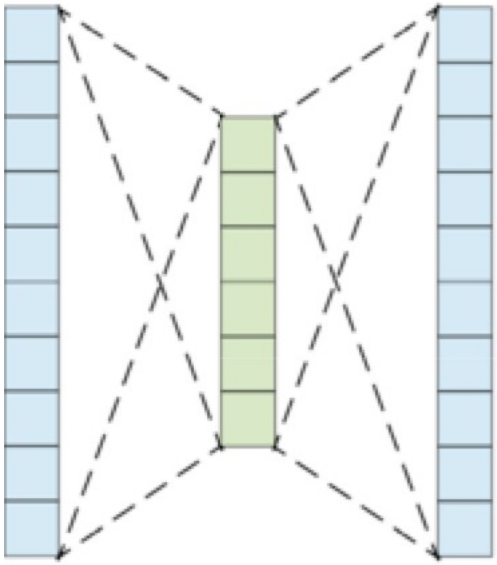
Przykład DBN [4]

Stacked autoencoders

- **Stack autoencoders**
  - **Wielowarstwowe sieci**
  - **Uczenie iteracyjne (warstwa ukryta w wytrenowanym autoenkoderze stanowi wejście dla kolejnej warstwy ukrytej)**
  - **Podczas uczenia i-tej warstwy latent, parametry wyższych warstw są blokowane w pierwszej fazie uczenia**

Stacked autoencoders

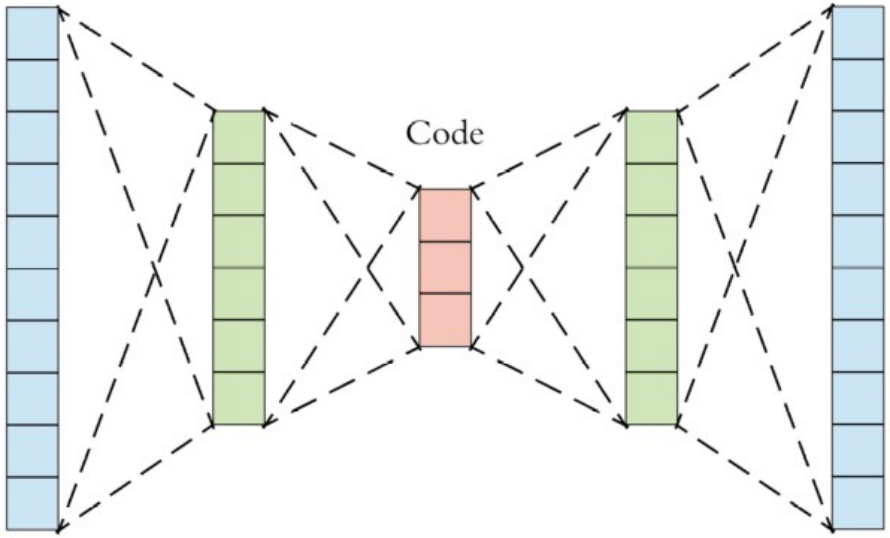
- **Uczenie autoenkoderów wielowarstwowych**
  - **Uczenie sieci z jedną warstwą ukrytą**



https://towardsdatascience.com/generating-images-with-autoencoders-77fd3a8dd368

Stacked autoencoders

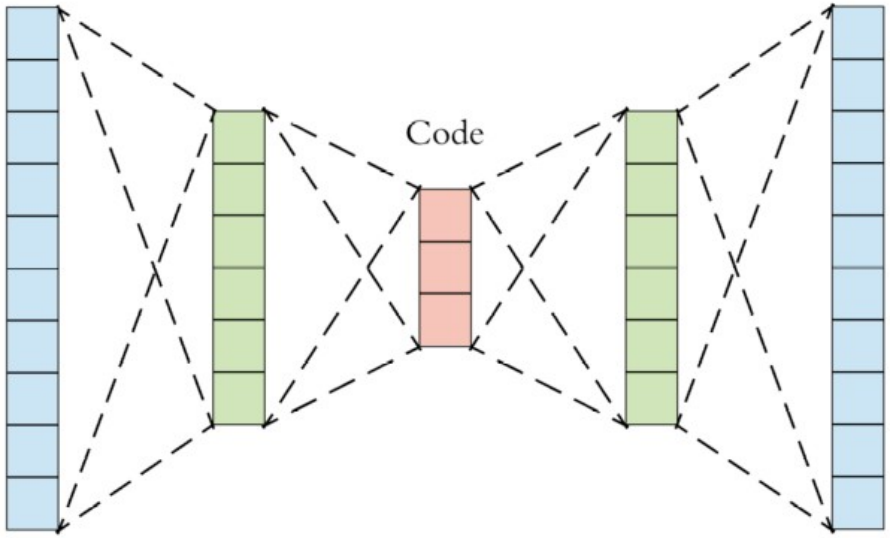
- **Uczenie autoenkoderów wielowarstwowych**
  - **Uczenie sieci z jedną warstwą ukrytą**
  - **Dodanie kolejnej warstwy ukrytej, zablokowanie wag warstwy już wyuczonej, uczenie nowej warstwy ukrytej**



https://towardsdatascience.com/generating-images-with-autoencoders-77fd3a8dd368

Stacked autoencoders

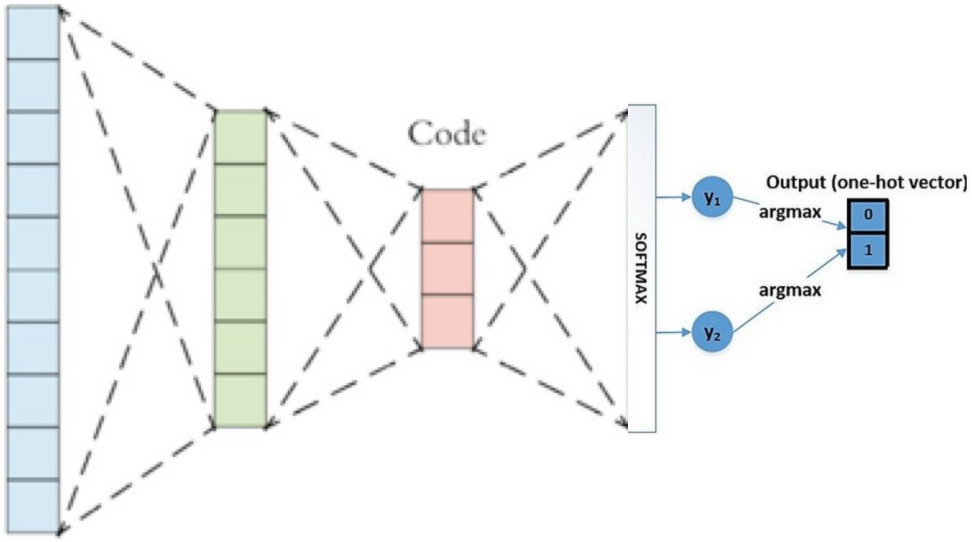
- **Uczenie autoenkoderów wielowarstwowych**
  - **Uczenie sieci z jedną warstwą ukrytą**
  - **Dodanie kolejnej warstwy ukrytej, zablokowanie wag warstwy już wyuczonej, uczenie nowej warstwy ukrytej**
  - **Odblokowanie wag i douczenie calej sieci**



https://towardsdatascience.com/generating-images-with-autoencoders-77fd3a8dd368

Stacked autoencoders

- **Zamiana dekodera na klasyfikator, douczenie go z zablokowanymi wagami enkodera a następnie wykonanie fine tuningu to tzw. uczenie pół nadzorowane**



https://towardsdatascience.com/generating-images-with-autoencoders-77fd3a8dd368, https://www.researchgate.net/profile/Babak_Bashari_Rad/publication/326914586/figure/fig1/AS:657671334150145@1533812473734/Proposed-neural-network-classifier-with-softmax-output-function-and-a-bias-unit.png]

Stacked autoencoders

- **Podczas uczenia klasyfikatora, w początkowej fazie wagi enkodera zostają zamrożone**
- **W przypadku klasyfikacji, podczas uczenia, jako funkcji błędu, używa się funkcji cross entropy**

$$H(p,q) = -\sum_{\forall y} p(y) log(q(y')) - \sum_{\forall y} (1-p(y)) log(1-q(y'))$$

Bibliografia

[1] https://skymind.ai/wiki/restricted-boltzmann-machine

[2] https://towardsdatascience.com/deep-learning-meets-physics-restricted-boltzmann-machines-part-i-6df5c4918c15

[3] https://pl.wikipedia.org/wiki/Iloczyn_diadyczny

[4] https://codeburst.io/deep-learning-deep-belief-network-fundamentals-d0dcfd80d7d4