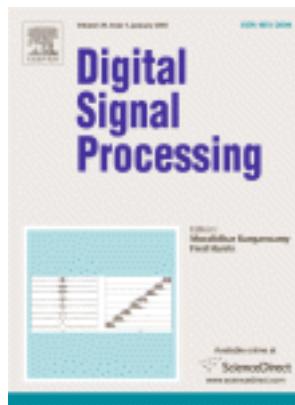
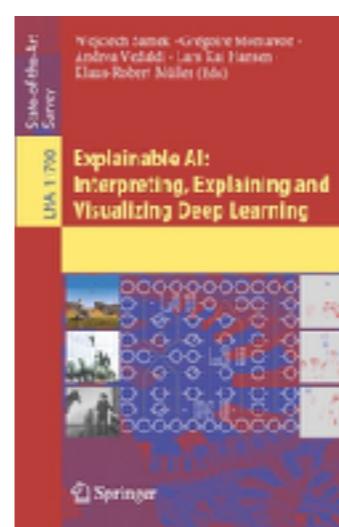


# Metody interpretacji i rozumienia decyzji podejmowanych przez głębokie sieci neuronowe



Methods for interpreting and understanding deep neural networks  
**Author:** Grégoire Montavon, Wojciech Samek, Klaus-Robert Müller  
**Publication:** Digital Signal Processing  
**Publisher:** Elsevier  
**Date:** February 2018  
*Copyright © 2018, Elsevier*



Explainable AI: Interpreting, Explaining and Visualizing Deep Learning  
**Editors:** Wojciech Samek, Grégoire Montavon, Andrea Vedaldi, Lars Kai Hansen, Klaus-Robert Müller

# Motywacja

- aby się upewnić, że problem jest prawidłowo reprezentowany i model nie skupia się lub wykorzystuje nieistotne szczegóły danych => dodatkowa walidacja modelu
- w medycynie, pojazdach autonomicznych i tam gdzie wymagane jest poleганie na modelu trzeba mieć pewność, że model wykorzystuje właściwe cechy
- aby zdobyć nowy wgląd w skomplikowane systemy w nauce

# Interpretacja post-hoc

- ten typ interpretacji zakłada, że mamy gotowy wytrenowany model klasyfikatora i chcemy zrozumieć co ten model przewiduje na podstawie zrozumienia typowych wzorców
- jest to w kontraste do budowania modeli, które w swojej strukturze zawierają element wyjaśniania => np. modele liniowe w statystce, drzewa decyzyjne, technika class-activation-mapping (CAM)
- techniki wyjaśniania niezależne od modelu, np. SHAP

# SHAP - Shapley Additive exPlanations

## Wyjaśnianie metodą addytywnych wartości Shapleya

- Metoda oparta jest o pomysł Lloyda Shapleya (1951), dotyczącego sprawiedliwego podziału zysku pomiędzy graczy będących w koalicji w grach kooperacyjnych.
- Do ML została zaadaptowana przez Scott M. Lundberg, Su-In Lee (2017)

Jak gracz (**cecha**) przykłada się do uzyskania wyniku (**predykcji**) ?

Wartości Shapleya mówią ile punktów zdobędziemy lub stracimy jeśli gra (**predykcja**) będzie bez udziału danego gracza (**cechy**).

# Wartości Shapleya

Załóżmy, że mamy 3 graczy A, B, C i chcemy obliczyć wartość gracza A:

- musimy oszacować jaki byłby wynik gry dla wszystkich kombinacji graczy (koalicji)
- Wartość Shapleya dla A jest ważoną sumą różnic w wynikach otrzymanych dla gry z udziałem A i bez udziału.
- Formalnie jest to określane jako:  
„znajdowanie brzegowego wkładu gracza uśrednionego po wszystkich możliwych sekwencjach, w których gracze dołączali do gry”

$$v_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \underbrace{\left[ f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S) \right]}_{\text{zysk z dodania } i \text{ do koalicji } S}$$

Notacja:  $|F|$  to rozmiar pełnej koalicji,  $S$  to podzbiór koalicji, w których nie grał gracz  $i$ ,  $|S|$  to rozmiar tego podzbioru.

$$\begin{aligned} v_A &= \omega_1(f(x_A, -, -, -) - f(-, -, -, -)) + \\ &\quad \omega_2(f(x_A, x_B, -) - f(-, x_B, -)) + \\ &\quad \omega_3(f(x_A, -, x_C) - f(-, -, x_C)) + \\ &\quad \omega_4(f(x_A, x_B, x_C) - f(-, x_B, x_C)) \end{aligned}$$

$f(x_A, -, -)$  oznacza wynik gry, w której nie grał B i C, tylko sam A

$f(-, -, -)$  oznacza wynik gry, w której nikt nie grał

u nas  $|S|$  wynosi odpowiednio 0, 1, 1, 2

zatem odpowiednio  $\omega_1 = 1/3, \omega_2 = 1/6, \omega_3 = 1/6, \omega_4 = 1/3$

# Wartości Shapleya w ML

- Niech wejściem do modelu jest wektor cech  $X$  o rozmiarze  $M$
- oznaczmy uproszczone wejście  $X'$ , w którym 1 na danej współrzędnej  $i$  oznacza, że model ma uwzględniać daną cechę  $i$  a 0, że ta cecha ma być ignorowana

$$v_i(f, x) = \sum_{z' \subseteq x'} \frac{|z'|! (|M| - |z'| - 1)!}{|M|!} [f_x(z') - f_x(z' \setminus i)]$$

Notacja:

- $|z'|$  to to liczba niezerowych współrzędnych w  $z'$ ,
- $z' \subseteq x'$  to wszystkie wektory  $z'$ , których niezerowe współrzędne są podzbiorem niezerowych współrzędnych w  $x'$ .
- Idea autorów metody SHAP zakłada, że: **predykcja modelu z usuniętą jedną lub więcej cechami jest przybliżana przez wartość oczekiwana po predykcjach z możliwymi wszelkimi wartościami wstawionymi w miejsce cech(y) usuniętych.**

$$f_x(z') = E[f(x)|z_S] \quad \text{gdzie } S \text{ jest zbiorem niezerowych indeksów w } z'$$

**Przykład:** mamy model z cechami A, B, C i chcemy wyjaśnić wpływ poszczególnych cech na uzyskany wynik  $f(5, 3, 10) = 7$

Aby obliczyć wartości Shapleya wprost dla poszczególnych cech powinniśmy wyliczyć wszystkie kombinacje  $f(x_A, x_B, x_C)$  np. :

$f(5, 3, -)$ : ustawiamy  $x_A = 5$  i  $x_B = 3$ , wyliczamy predykcje dla wszystkich wartości  $x_C$  jakie mamy w zbiorze treningowym i uśredniamy uzyskane dla tych przykładów predykcje.

$f(-, -, 10)$ : ustawiamy  $x_C = 10$ , sprawdzamy jakie możliwe wartości przyjmuje w zbiorze treningowym  $x_A$  i jakie  $x_B$ . Dla wszystkich kombinacji wartości  $(x_A, x_C)$  i ustalonego  $x_C = 10$  wyliczamy predykcję i uśredniamy.

$f(-, -, -)$ : wymaga wyliczenia predykcji dla wszystkich możliwych kombinacji wartości cech w zbiorze treningowym

Widać, że w takim bezpośrednim podejściu do liczenia wartości Shapleya złożoność obliczeniowa bardzo szybko rośnie wraz z liczbą cech i ich możliwych wartości. Autorzy metody SHAP zaproponowali kilka aproksymacji umożliwiających efektywne szacowanie tych wartości

# Wartości Shapleya w ML

- Aby uprościć obliczenia autorzy SHAP zaproponowali następujące przybliżenia:

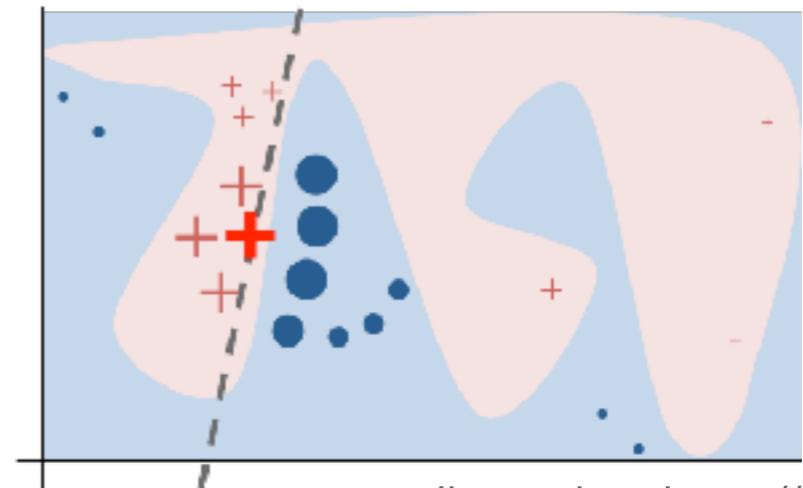
$$f_x(z') = E[f(z) | z_S]$$

$$= E_{z_{\bar{S}}|z_S}[f(z)]$$

$$\approx E_{z_{\bar{S}}}[f(z)]$$

$$\approx f([z_S, E[z_{\bar{S}}]])$$

uśredniamy po przykładach dla których  $z'$  miało wyzerowane indeksy  
zakładając niezależność cech  
zakładając liniowość modelu



Ilustracja z: <https://arxiv.org/pdf/1705.07874.pdf>

# Przykład w notebooku

<https://colab.research.google.com/drive/1c9usDnY2116FsDid3iaHYhr>

# Deep Visualization Toolbox

[yosinski.com/deepvis](http://yosinski.com/deepvis)

#deepvis



Jason Yosinski



Jeff Clune



Anh Nguyen



Thomas Fuchs



Hod Lipson



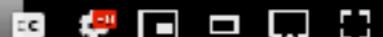
Cornell University



UNIVERSITY  
of WYOMING



Jet Propulsion Laboratory  
California Institute of Technology



<http://yosinski.com/deepvis>

# Definicje

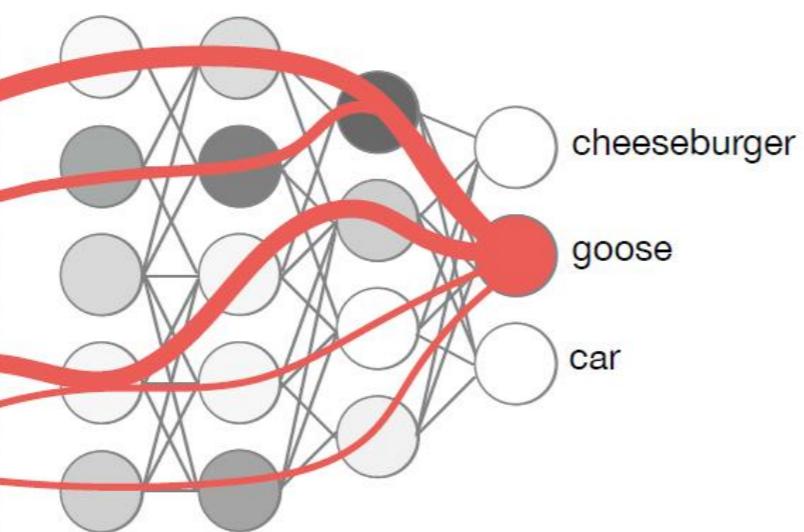
- **interpretacja** to mapowanie abstrakcyjnego pojęcia (przewidywanej klasy) do dziedziny, w której człowiek może zrozumieć sens tej predykcji. Taką dziedziną mogą być np. obrazy, teksty itp.
- **wyjaśnienie** to kolekcja cech w dziedzinie zrozumiałej dla człowieka, które przyczyniły się do danej predykcji.  
Cechom tym można też przypisać pewne wagi.
  - przykładem wyjaśnienia może być mapa kolorująca piksele w zależności od ich wkładu do decyzji
  - w klasyfikacjach związanych z językiem może to być podświetlenie słów i zwrotów przyczyniających się do decyzji
- Ten nacisk na rozumienie i interpretację przez człowieka jest też związany z aspektem prawnym „przypisania odpowiedzialności” i „prawa do wyjaśnienia”

## Interpretacja modelu

- znajdź prototypowy przykład dla danej klasy
- znajdź wzorzec maksymalizujący aktywność danego neuronu



simple regularizer  
(Simonyan et al. 2013)



$$\max_{x \in \mathcal{X}} p_\theta(\omega_c | x) + \lambda \Omega(x)$$

## Wyjaśnianie decyzji

- dlaczego model wykonał taką klasyfikację
- weryfikacja, że model działa zgodnie z naszą intuicją i rozumieniem problemu



ML blackbox

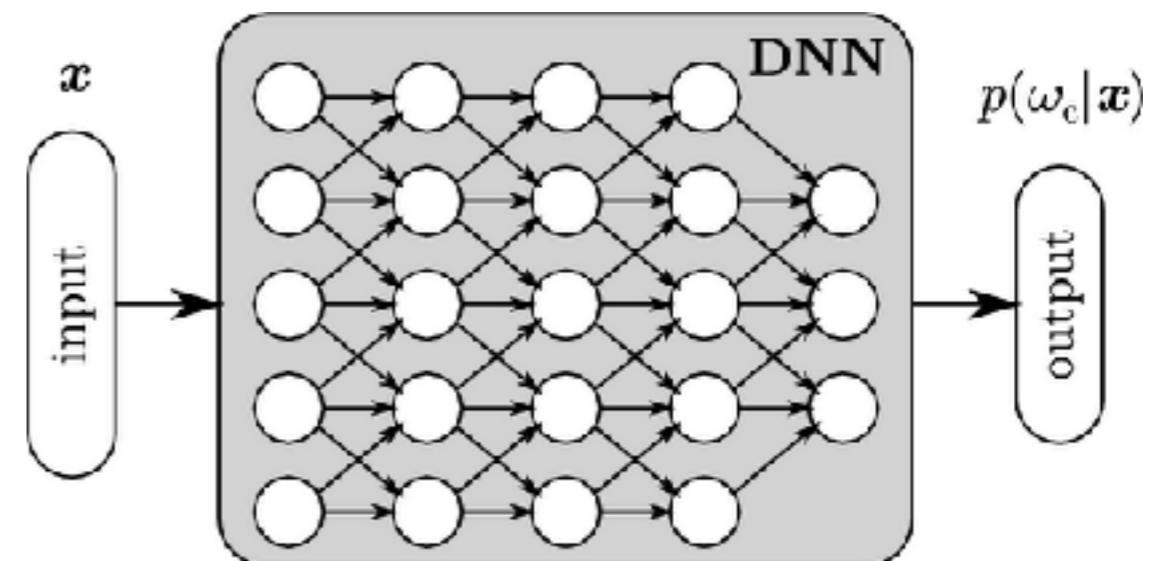
decision  
*it's a  
shark*

# Popularne podejścia do wizualizacji cech

- Dekonwolucja:
  - <https://www.matthewzeiler.com/mattzeiler/adaptivedeconvolutional.pdf>
  - Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 818–833. Springer, Cham (2014).  
[https://doi.org/10.1007/978-3-319-10590-1\\_53](https://doi.org/10.1007/978-3-319-10590-1_53)
- Guided backpropagation:
  - Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.: Striving for simplicity: the all convolutional net. In: ICLR Workshop (2015)  
<https://arxiv.org/pdf/1412.6806.pdf>

# Interpretacja sieci głębokiej

- Neurony w sieci łącznie tworzą skomplikowane nieliniowe mapowanie z przestrzeni cech do przestrzeni klas
- neurony wyjściowe odpowiadają abstrakcyjnym pojęciom
- przestrzeń wejściowa jest interpretowalna (obrazy są interpretowalne dla człowieka)
- skupimy się teraz na tworzeniu prototypu w przestrzeni wejść, który reprezentuje wyuczone abstrakcyjne pojęcie
  - stworzymy go w ramach podejścia maksymalizacji aktywności



# Maksymalizacja aktywności (MA)

- Poszukiwanie takiego wzorca wejściowego, który maksymalizuje wyjście dla pewnej klasy

$$x^* = \max_x \log p(\omega_C | x) - \lambda \|x\|^2$$

- $(\omega_c)_c$  zbiór klas
- $p(\omega_c | x)$  - prawdopodobieństwo przynależności do klasy zwracane przez neuron wyjściowy.
- prawdopodobieństwa klas można maksymalizować metodami gradientowymi
- ta metoda zastosowana od obrazów zwraca obrazki w większości szare z kilkoma najważniejszymi pikslami lub krawędziami
- prototypy choć maksymalizują wyjście to nie wyglądają dla człowieka naturalnie



# Ulepszanie MA

- zamiast regularyzacji  $\ell_2$  można użyć innych, zawierających pewną wiedzę o danych, np. ich rozkład prawdopodobieństwa

$$\max_x \log p(\omega_c | x) + \log p(x)$$

- ta regularyzacja prowadzi przez regułę Byesa do

$$\max_x \log p(\omega_c | x)p(x) = \max_x \log p(x | \omega_c)$$

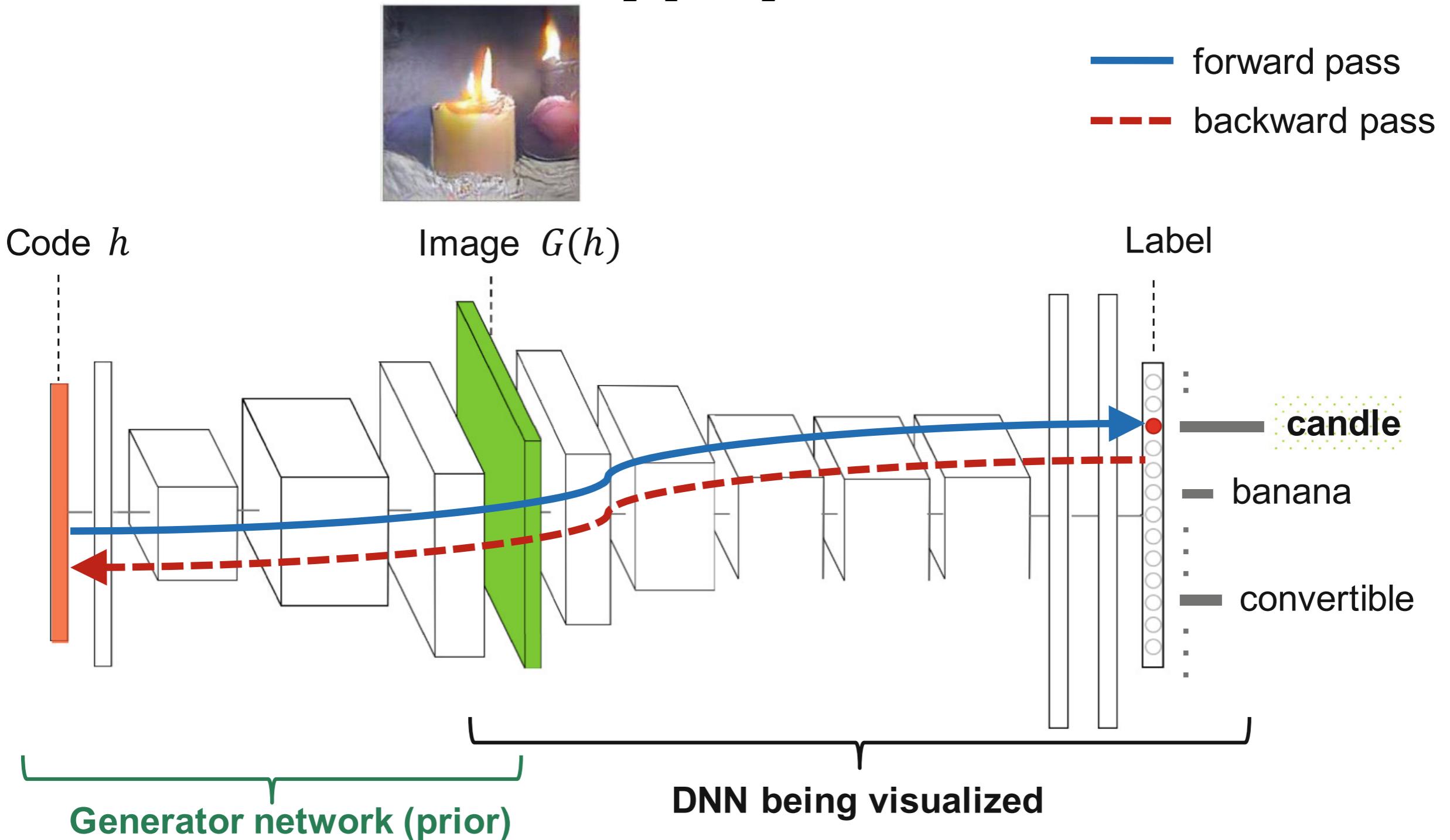
czyli do wzorca w ciągu uczącym, który jest najbardziej charakterystyczny dla swojej klasy

- opublikowanych pomysłów na regularyzację jest więcej

# Algorytm MA w przestrzeni abstrakcyjnych kodów

- modele generatywne - metoda ucznia bez nauczyciela. Modele te nie dają wprost funkcji gęstości  $p(x)$ , ale potrafią z niej próbkować:
  - pobierają próbkę z jakiegoś prostego rozkładu  $q(z) \sim \mathcal{N}(0, I)$  zdefiniowanego w abstrakcyjnej przestrzeni kodów  $\mathcal{Z}$
  - Stosują funkcję dekodującą  $g: \mathcal{Z} \rightarrow \mathcal{X}$ , która mapuje do oryginalnej przestrzeni wejść  $\mathcal{X}$
- Przykładem takich modeli są GAN generative adversarial network

# Algorytm MA w przestrzeni abstrakcyjnych kodów



# Algorytm MA w przestrzeni abstrakcyjnych kodów

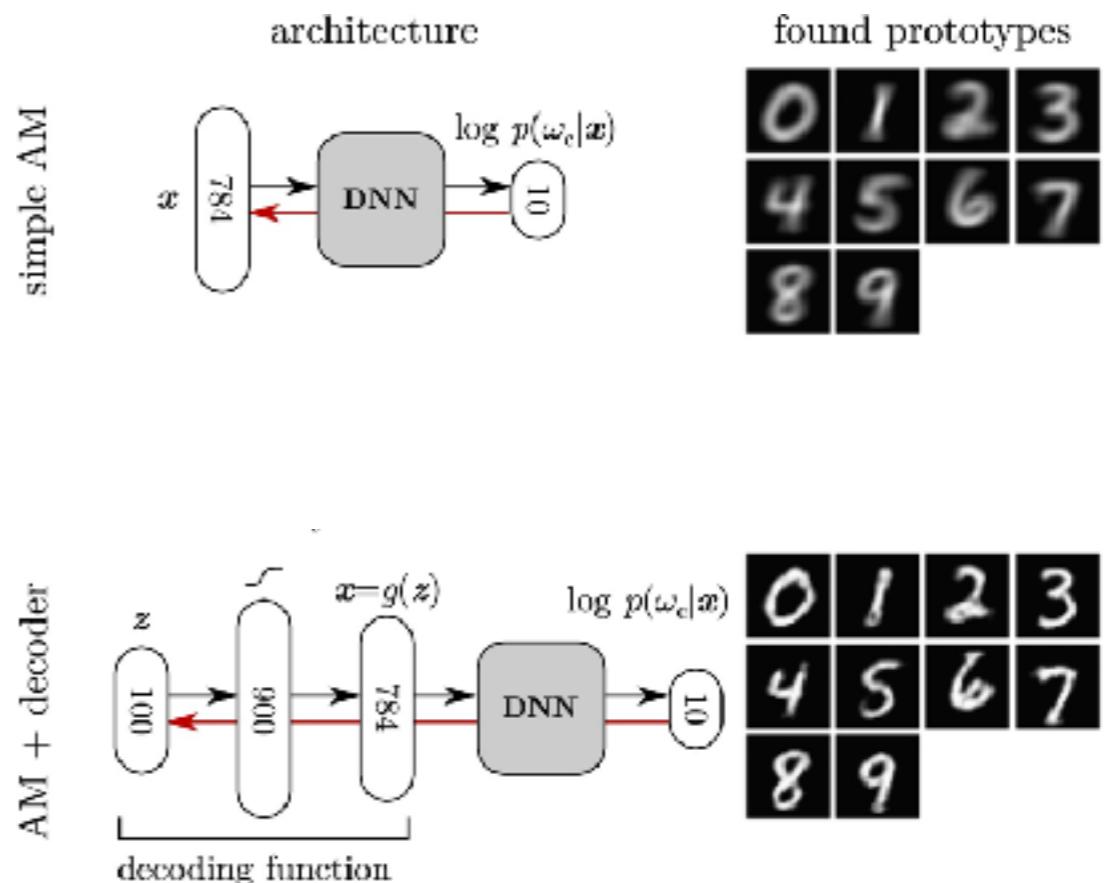
- Nguyen zaproponował wbudowanie takiego generatora wprost do algorytmu MA; jego problem optymalizacyjny jest następujący:

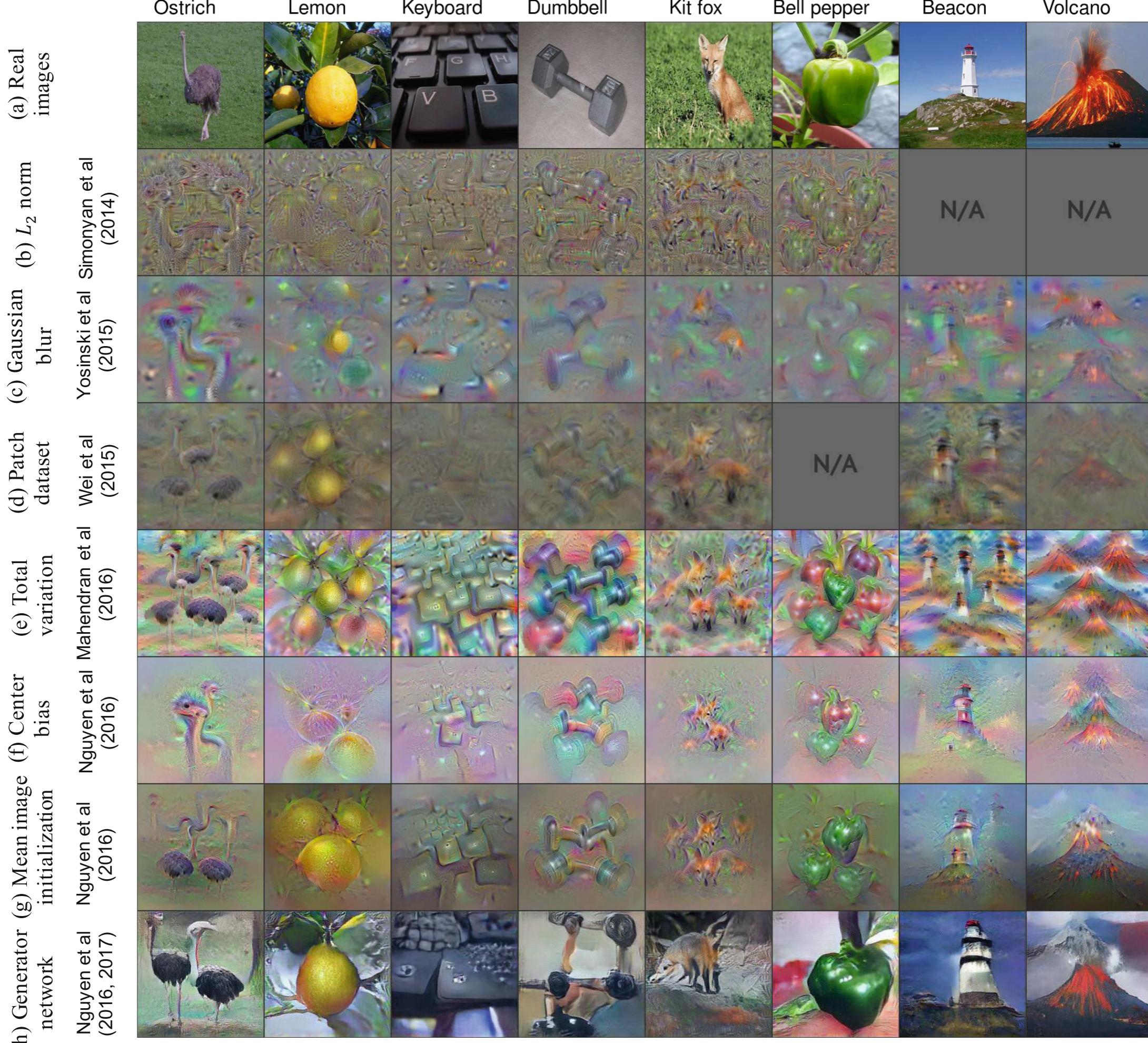
$$\max_{z \in \mathcal{Z}} \log p(\omega_c \mid g(z)) - \lambda \|z\|^2$$

- po znalezieniu optymalnego  $z^*$  prototyp dla klasy  $\omega_c$  jest znajdowany jako  $x^* = g(z^*)$
- dla normalnego rozkładu kodów  $q(z)$  człon regularyzacyjny jest równoważny  $\log q(z)$ , czyli faworyzuje kody o wysokim prawdopodobieństwie

# Interpretacja klasyfikacji zbioru MNIST

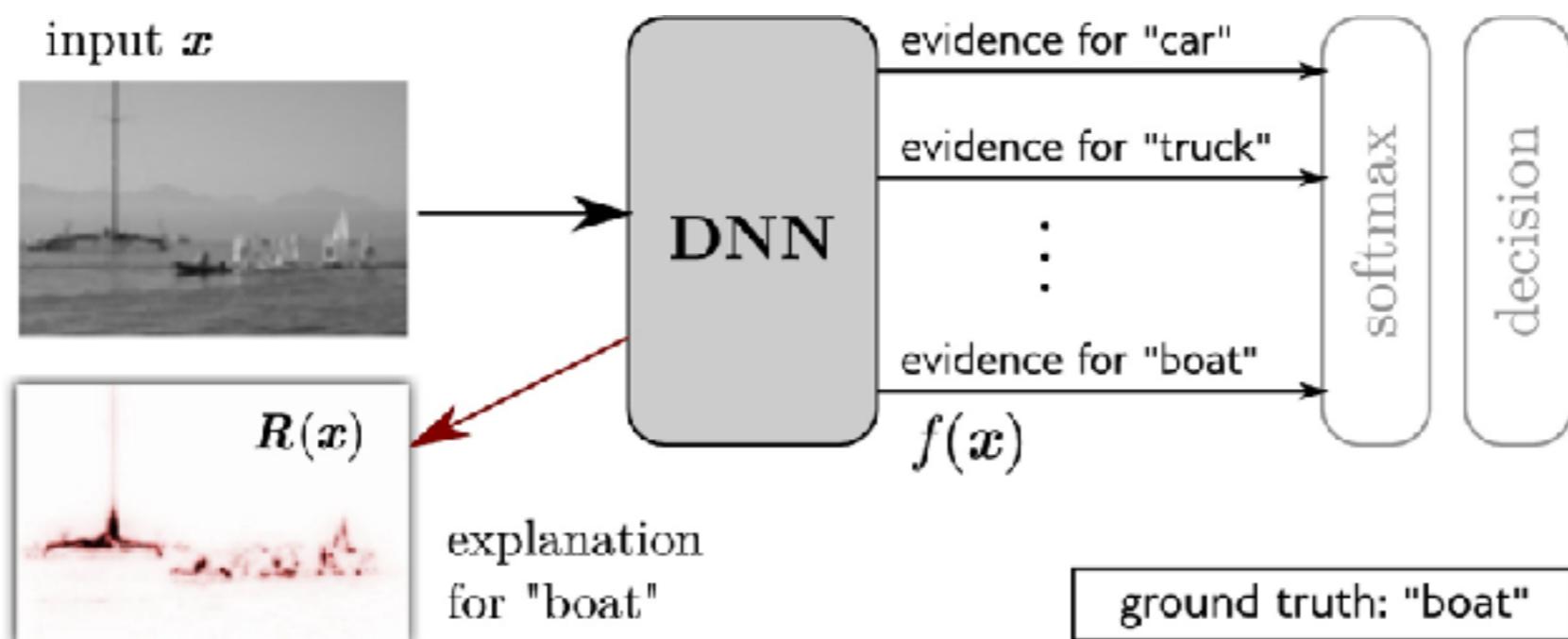
- prosta regularyzacja normą  $l_2$
- z modelem generatywnym w postaci dwuwarstwowego dekodera, regularyzacja normą  $l_2$ , przy czym w funkcji celu wyrażenie było postaci  $\lambda\|z - \bar{z}\|^2$ ,  $\bar{z}$  oznacza średni kod dla klasy  $\omega_c$





# Bardziej skomplikowane klasy - wyjaśnianie DNN

- Pytanie: „Jakie cechy wzorca  $x$  powodują, że jest on dobrym reprezentantem klasy  $\omega_c$ ?
- głęboka sieć neuronowa produkuje wyjście  $f(x)$
- patrzymy na wzorzec  $x$  jako na zbiór cech  $(x_i)_{i=1}^d$
- dla każdej cechy przypisujemy rangę  $R_i$ , która mówi jak istotna jest cecha  $x_i$  dla wyjaśnienia  $f(x)$



# Analiza czułości

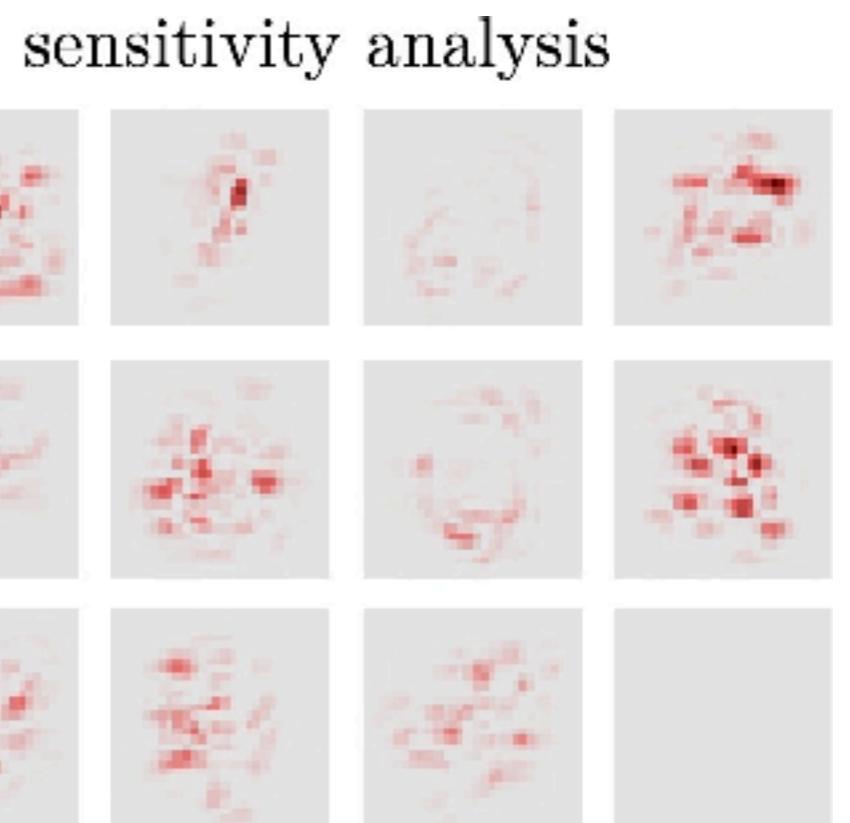
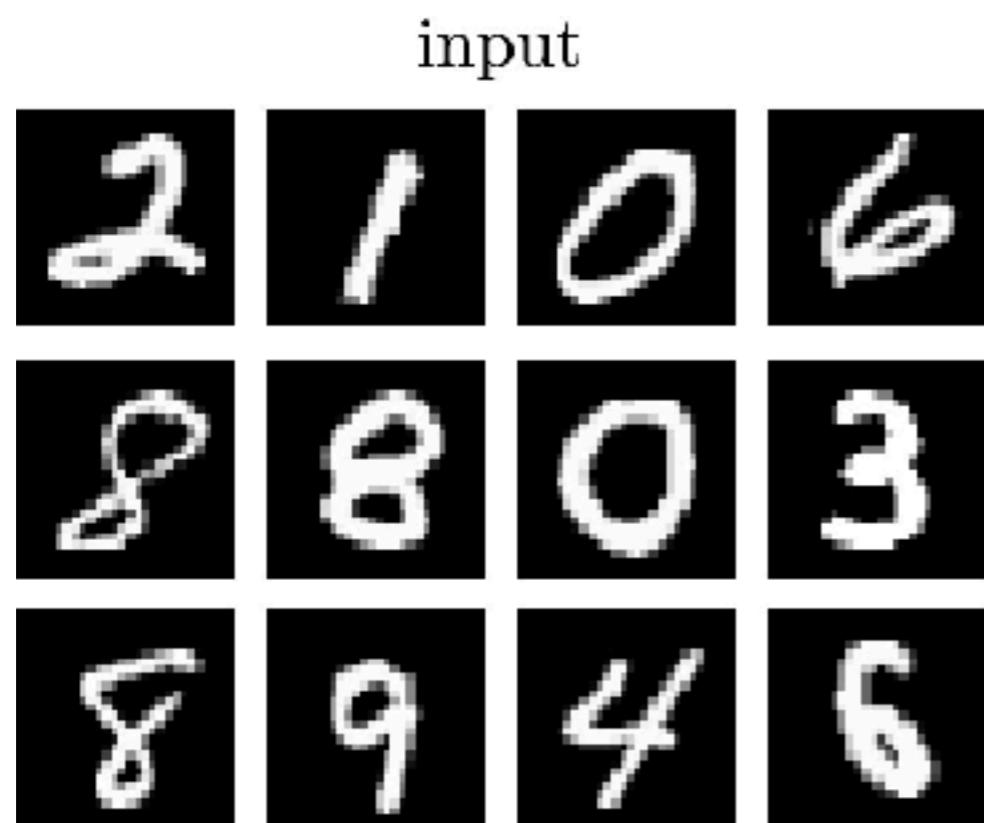
- $R_i(x) = \left( \frac{\partial f}{\partial x_i} \right)^2$ ,  
gradient jest obliczany w punkcie  $x$ .

- ta analiza skupia się na zmianach funkcji, a nie na jej wartościach

$$\sum_{i=1}^d R_i(x) = \|\nabla f(x)\|^2$$

- w tym sensie najbardziej istotne cechy to te, które powodują największe zmiany funkcji  $f(x)$
- zaletą jest to, że gradienty można wyliczać w ramach algorytmu wstecznej propagacji

- mapa termiczna wskazuje, które piksele powodują, że cyfra należy do klasy docelowej *bardziej / mniej*
- nie wskazuje, co sprawia, że cyfra należy do tej klasy.



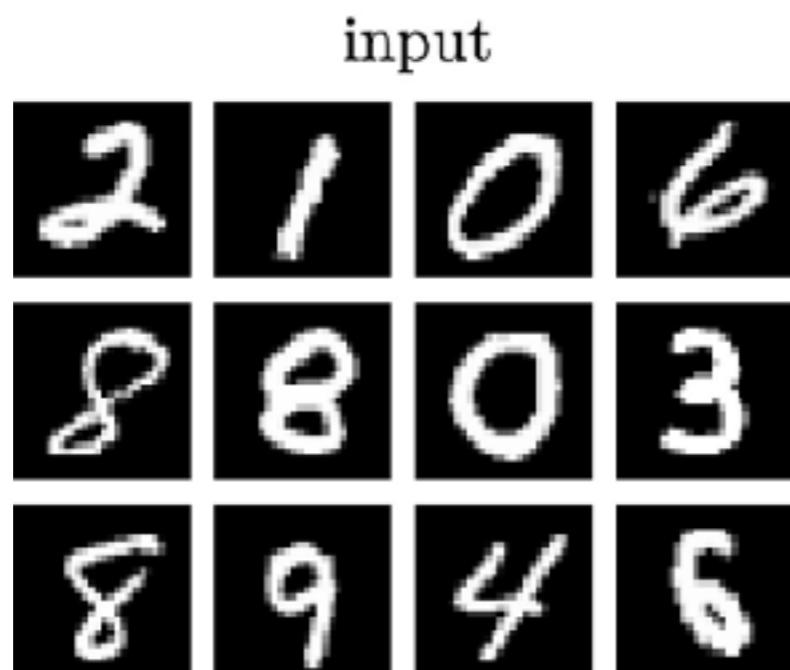
# Proste rozwiniecie Taylora

- Rozwiniecie Taylora jest w rozważanym kontekście zapisaniem  $f(x)$  jako sumy rang istotności
- rangi są otrzymywane jako rozwiniecia  $f(x)$  pierwszego rzędu wokół punktu  $\tilde{x}$  dla którego  $f(\tilde{x}) = 0$ , zatem

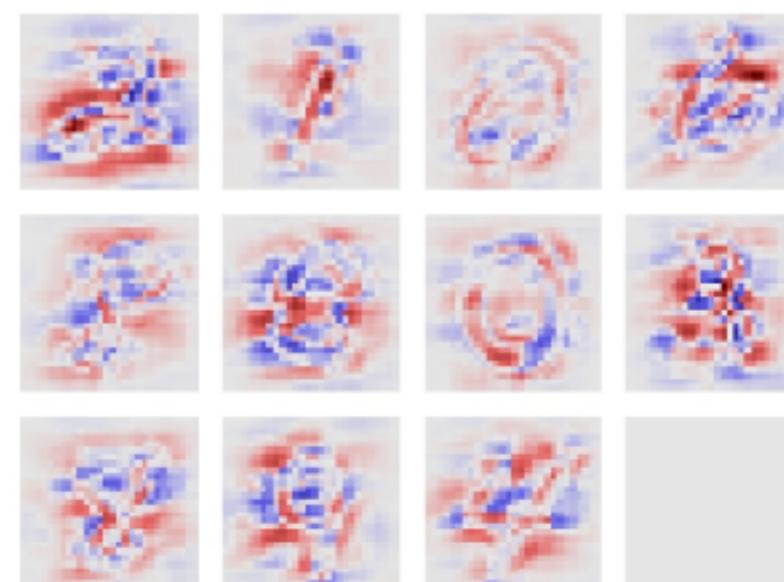
$$f(x) = \sum_{i=1}^d R_i(x) + O(x x^\top)$$

gdzie

$$R_i(x) = \frac{\partial f}{\partial x_i} \Big|_{x=\tilde{x}} \cdot (\tilde{x}_i - x_i)$$



simple Taylor  
decomposition

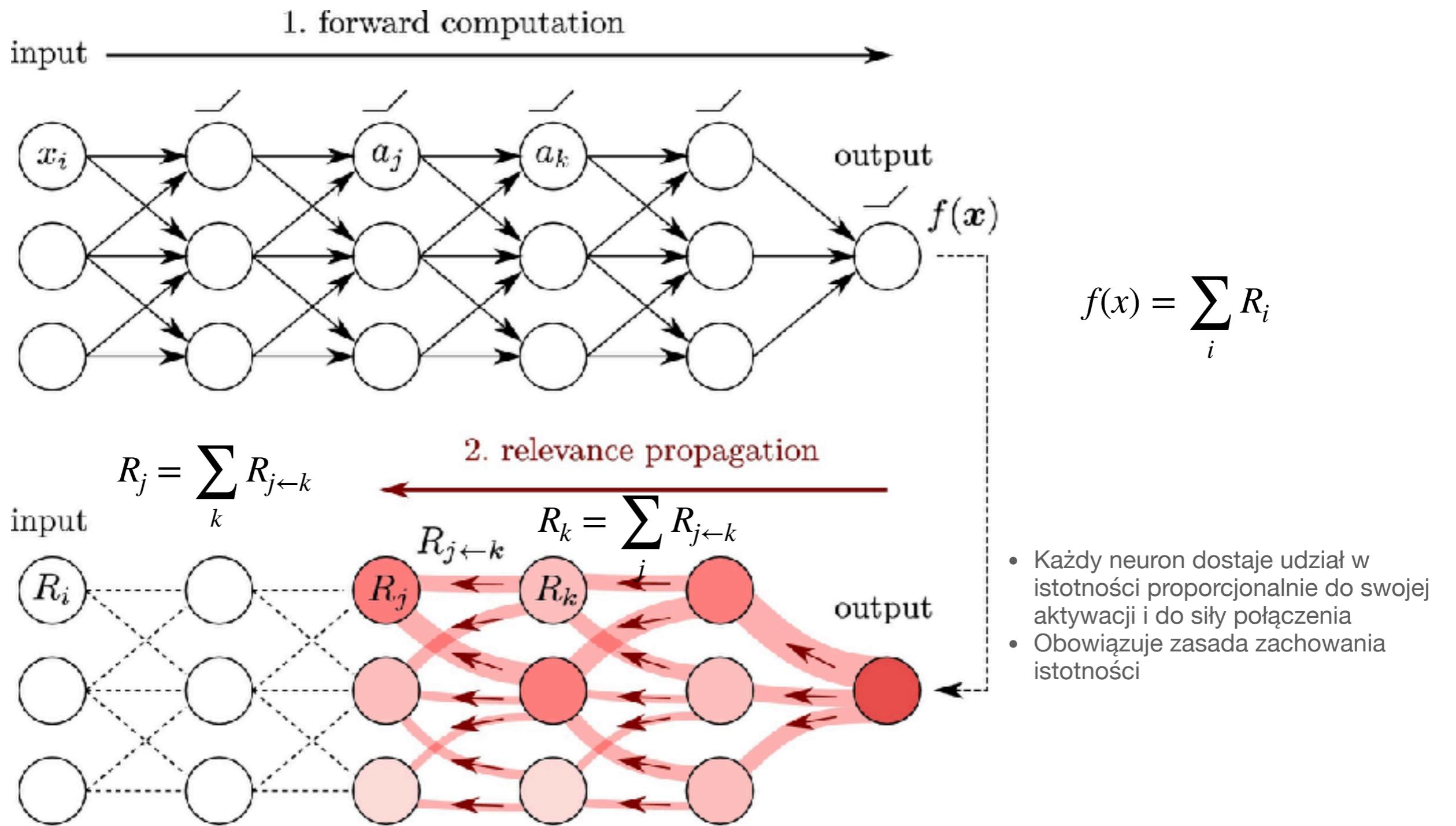


# Techniki propagacji wstecznej

- ogólny pomysł polega na wykorzystaniu struktury grafu tworzącego sieć:

Zaczynamy od wyjścia sieci. Następnie poruszamy się po grafie w odwrotnym kierunku, stopniowo mapując prognozę na niższe warstwy. Procedura kończy się po osiągnięciu wejścia sieci.

# Layer-wise relevance propagation (LRP) -warstwowa propagacja istotności (Bach et al. 2015)



# Zasady propagowania istotności

- niech aktywność neuronu będzie

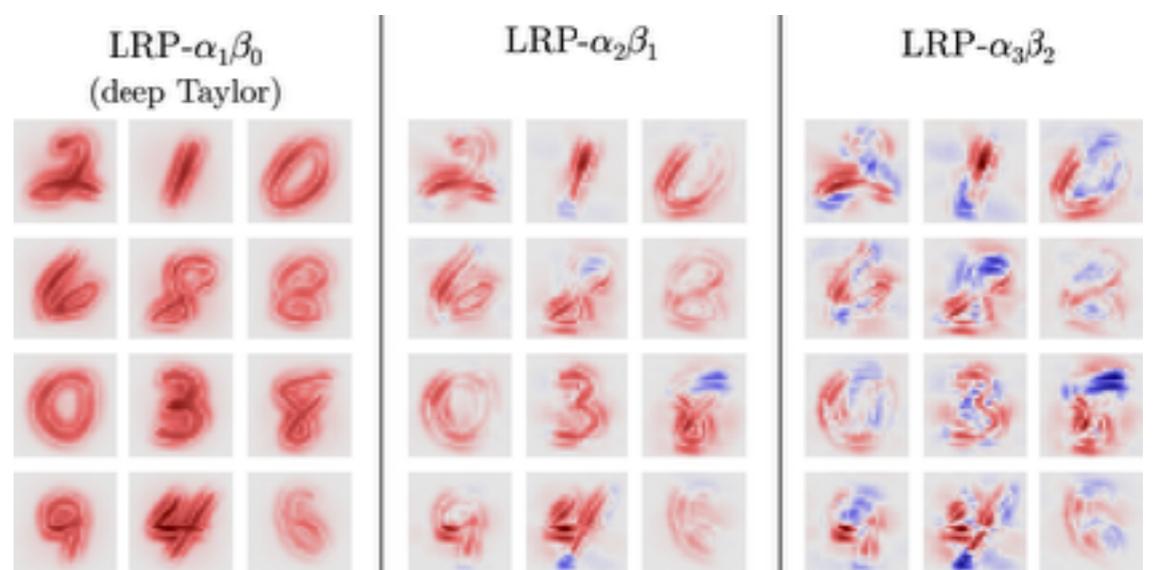
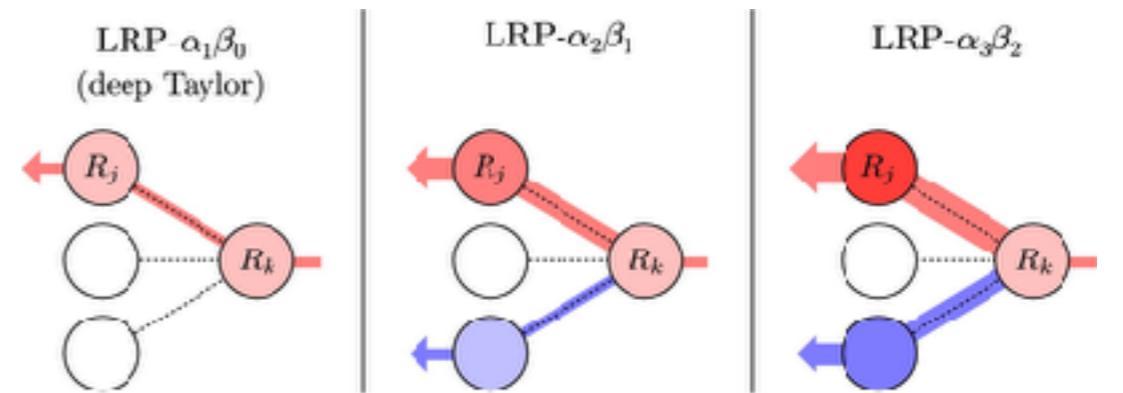
$$a_k = \sigma \left( \sum_j a_j w_{jk} + b_k \right)$$

- zasada  $\alpha\beta$ :

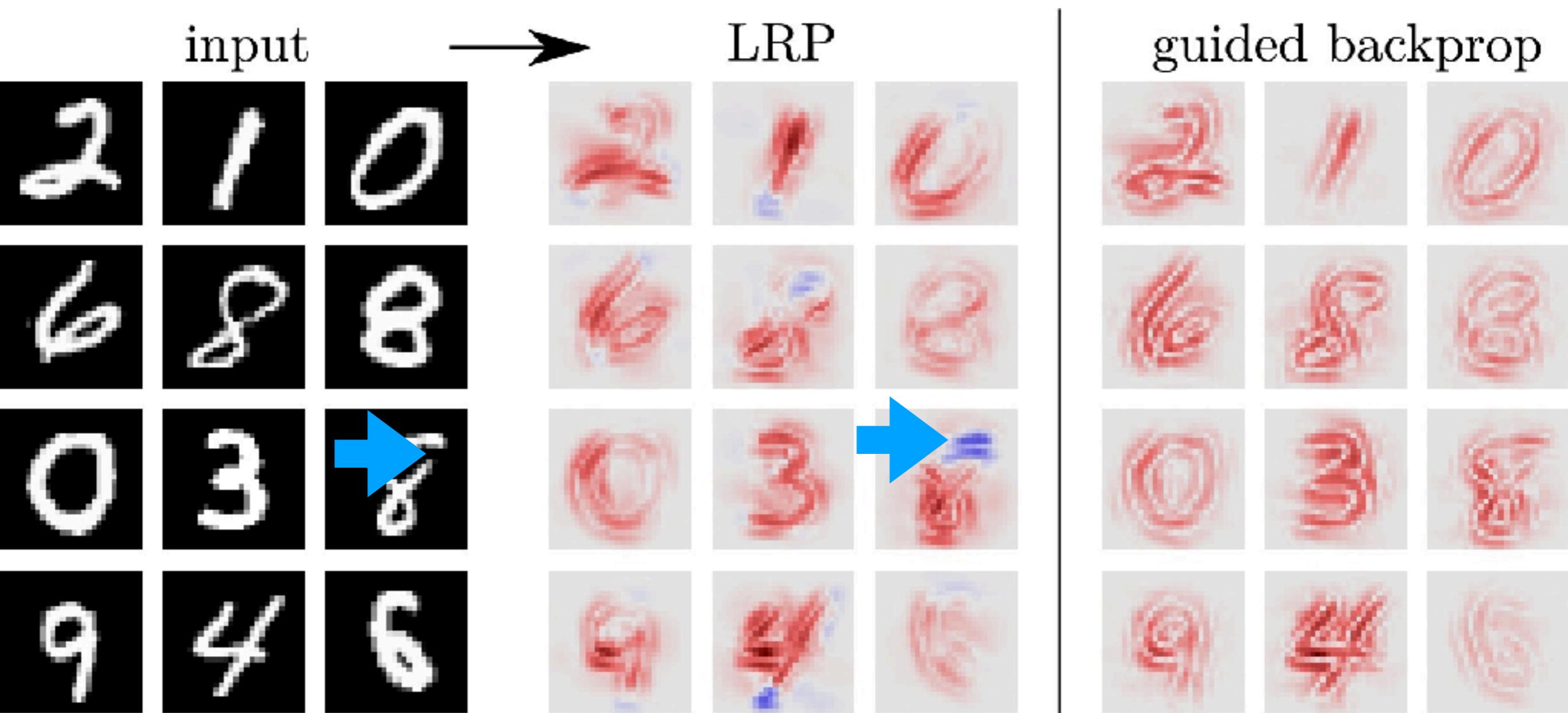
$$R_j = \sum_k \left( \alpha \frac{a_j w_{jk}^+}{\sum_j a_j w_{jk}^+} - \beta \frac{a_j w_{jk}^-}{\sum_j a_j w_{jk}^-} \right) R_k,$$

- z dodatkowym warunkiem:  $\alpha - \beta = 1$  i  $\beta > 0$

- parametry  $\alpha, \beta$  można wybierać różnie
- suma istotności jest stała  
 $(w_{jk})_j = (1, 0, -1)$

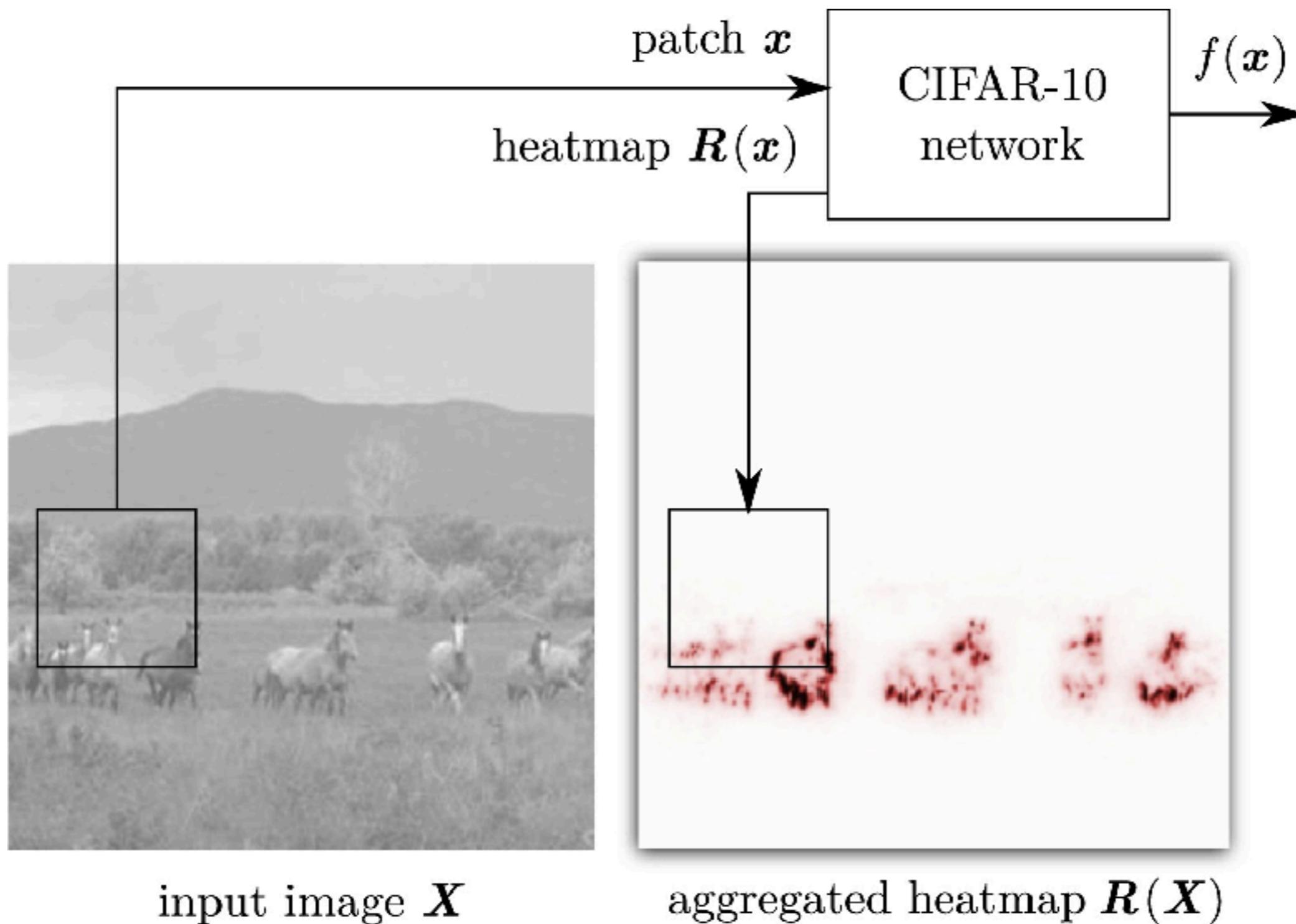


# Layer-wise relevance propagation (LRP) -warstwowa propagacja istotności (Bach et al. 2015)



przykładowy kod do obliczania LRP można znaleźć na  
<http://heatmapping.org/tutorial>

# Skanowanie dużych obrazków za pomocą okienka



# **Przykłady zastosowań**

# Walidacja modelu - ocena sensu przez człowieka

(a)

SVM/BoW classifier

target class sci.space.

on a roller coaster ride than others. The mental part is usually induced by a lack of clear indication of which way is up or down, ie: the Shuttle is normally oriented with its cargo bay pointed towards Earth, so the Earth (or ground) is "above" the head of the astronauts. About 50% of the astronauts experience some form of motion sickness, and NASA has done numerous tests in

CNN/word2vec classifier

on a roller coaster ride than others. The mental part is usually induced by a lack of clear indication of which way is up or down, ie: the Shuttle is normally oriented with its cargo bay pointed towards Earth, so the Earth (or ground) is "above" the head of the astronauts. About 50% of the astronauts experience some form of motion sickness, and NASA has done numerous tests in

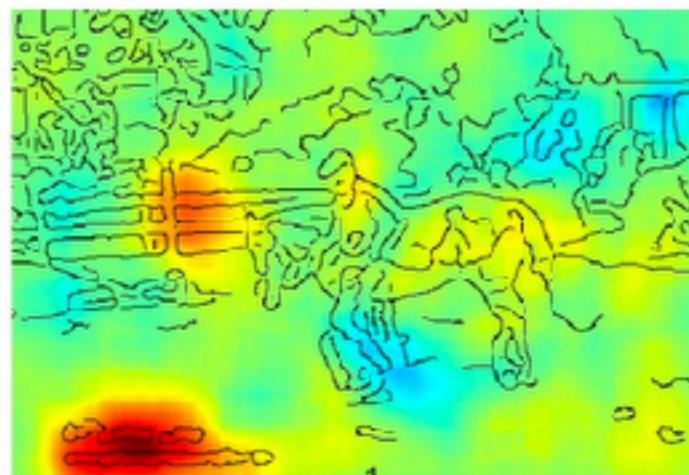
Based on Arras et al. (2016) "What is relevant in a text document? an interpretable ML approach"

(b)

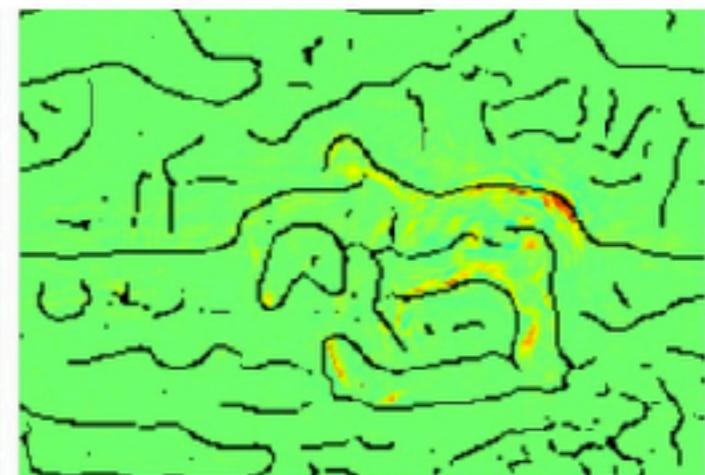
input image



"horse" classification by  
Fisher vectors



"horse" classification by  
Deep neural networks



Based on Lapuschkin et al. (2016) "Analyzing classifiers: Fisher vectors and deep neural nets"

Oba klasyfikatory mają podobną skuteczność na zdjęciach koni.  
Niespodziewane użycie tagu copyright do klasyfikacji i fragmentu ogrodze

# Zastosowania do uzyskiwania wglądu w problemy naukowe: mapowanie sekwencji DNA na miejsca wiążania

Alipanahi i in. wytrenował sieć konwolucyjną do mapowania sekwencji DNA na miejsca wiążania białka.

Za jej pomocą testowano jakie nukleotydy z tej sekwencji są najbardziej istotne dla wyjaśnienia obecności tych miejsc wiążania.

Wykorzystali analizę opartą na zaburzeniach, w której mierzy się istotność każdego nukleotydu na podstawie wpływu mutacji na prognozę sieci neuronowej.

(c) sequence 1 (true positive)

... T I G G I C C C G T A A A G I T I A G T T T I C A C G T T I G A C I G ...

sequence 2 (false positive)

... C I C G I A C I A G G G C A C I T I A T I T I C A C G T T I G A C I A ...

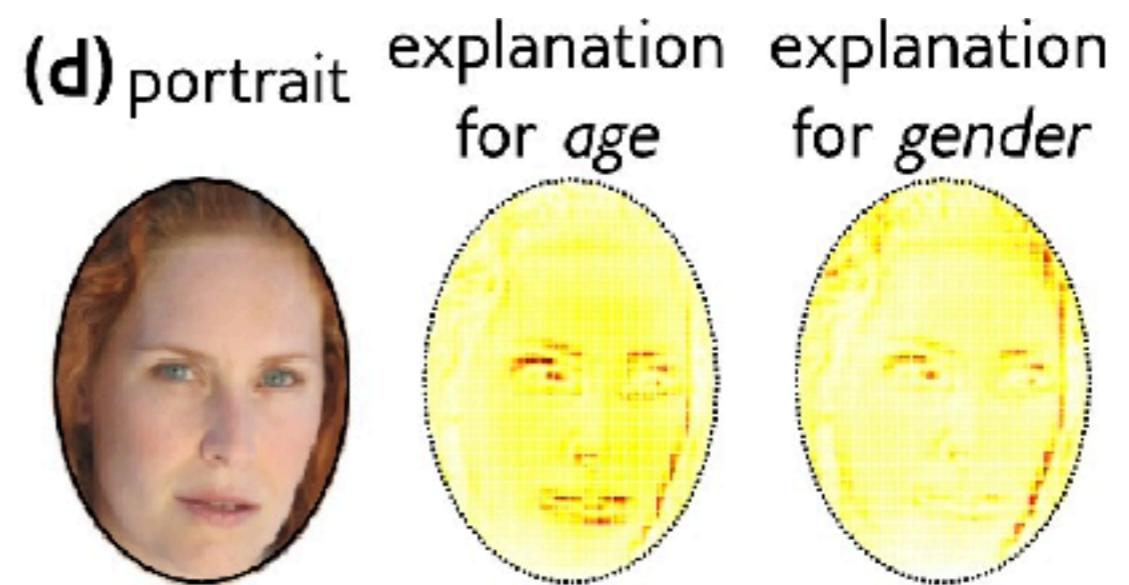
sequence 3 (false negative)

... G I C G I C I A G I A G I T I A C I A G T I A C T C I G T I A I G T G I T ...

Adapted from Vidovic et al. (2016) "Feature importance measure for non-linear learning algorithms"

# Zastosowania do uzyskiwania wglądu w problemy naukowe

- Techniki wyjaśniania mają również potencjalne zastosowanie w analizie obrazów twarzy.
- Ich bezpośrednia interpretacja pod kątem rzeczywistych cech obrazu wejściowego może być trudna.
- Arbabzadah i in. zastosował technikę LRP, aby zidentyfikować, które piksele na danym obrazie są odpowiedzialne za wyjaśnienie, na przykład, atrybutów wieku i płci.



Based on Arbabzadah et al. (2016)  
"Identifying individual facial expressions  
by deconstructing a neural network"

**Zastosowania z  
naszego podwórka :)**

# Zastosowania DNN do analizy EEG

Jedno z pierwszych podejść:

♦ Human Brain Mapping 38:5391–5420 (2017) ♦

**Deep Learning With Convolutional Neural Networks for EEG Decoding and Visualization**

Robin Tibor Schirrmeister  <sup>1,2,\*</sup>, Jost Tobias Springenberg, <sup>2,3</sup>  
Lukas Dominique Josef Fiederer  <sup>1,2,4</sup>, Martin Glasstetter, <sup>1,2</sup>  
Katharina Eggensperger, <sup>2,5</sup> Michael Tangermann, <sup>2,6</sup> Frank Hutter, <sup>2,5</sup>  
Wolfram Burgard, <sup>2,7</sup> and Tonio Ball 

# Zastosowania DNN do analizy EEG

na podstawie:

Agnieszka Wójtowicz

Nr albumu: 406063

**Wykorzystanie techniki LRP do  
wizualizacji i interpretacji sieci  
neuronowej wytrenowanej do  
klasyfikacji sygnału EEG**

Praca licencjacka

na kierunku zastosowania fizyki w biologii i medycynie

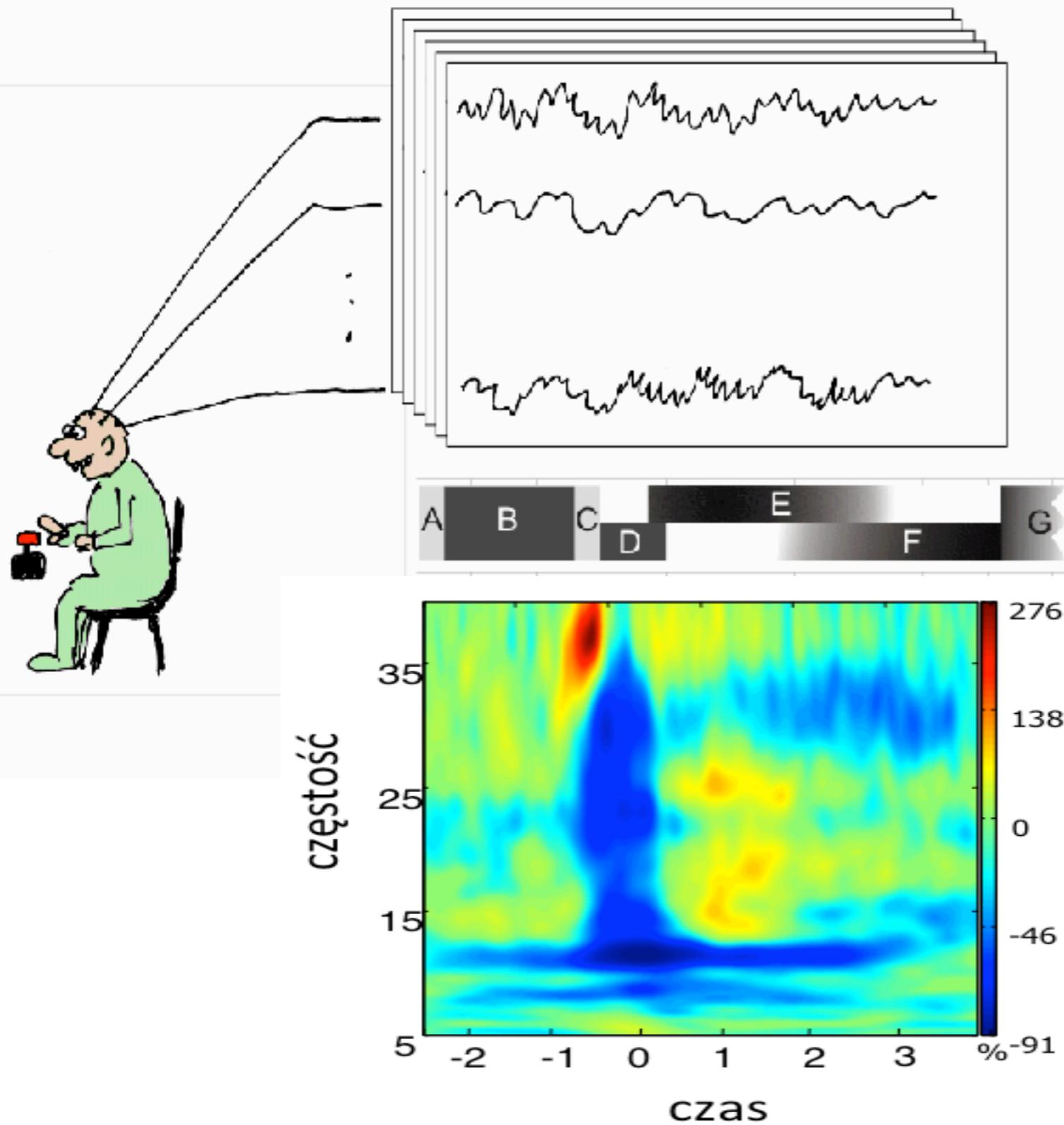
# Ogólnie

- Przykład zastosowania sieci głębokich do wytworzenia cech użytecznych do dekodowania sygnału EEG
- Demonstracja cech wytworzonych przez sieć
- Porównanie do cech opracowanych w tej dziedzinie przez ekspertów

# Reprezentacja danych wejściowych

- Sygnał EEG jest zazwyczaj wielokanałowy
- Kanały nie są w pełni niezależne - mają strukturę korelacyjną w przestrzeni
- W ramach danego kanału występuje struktura korelacji czasowych
- Istotne informacje niesione są też przez różne pasma częstotliwości
- W klasycznym podejściu wykorzystujemy filtry przestrzenne, filtry częstotliwościowe, analizę morfologiczną (kształtu)

# Paradygmat eksperymentów ERD/ERS



Przykładowa mapa zjawiska synchronizacji i desynchronizacji sygnału EEG w przestrzeni czas-częstość dla sygnału zarejestrowanego przez elektrodę C3.

Osoba badana wykonywała szybki ruch palcem w momencie oznaczonym 0.

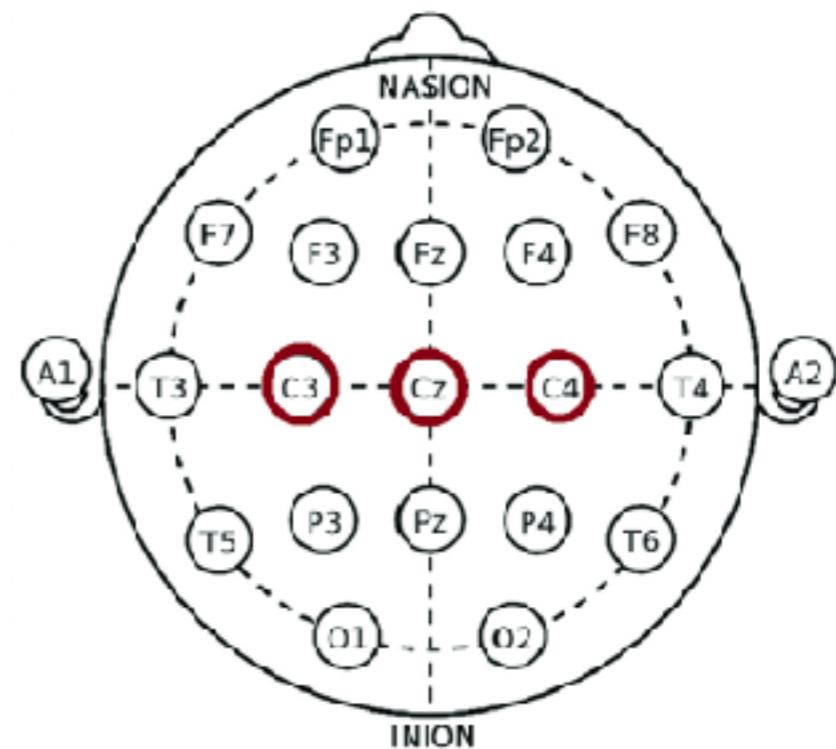
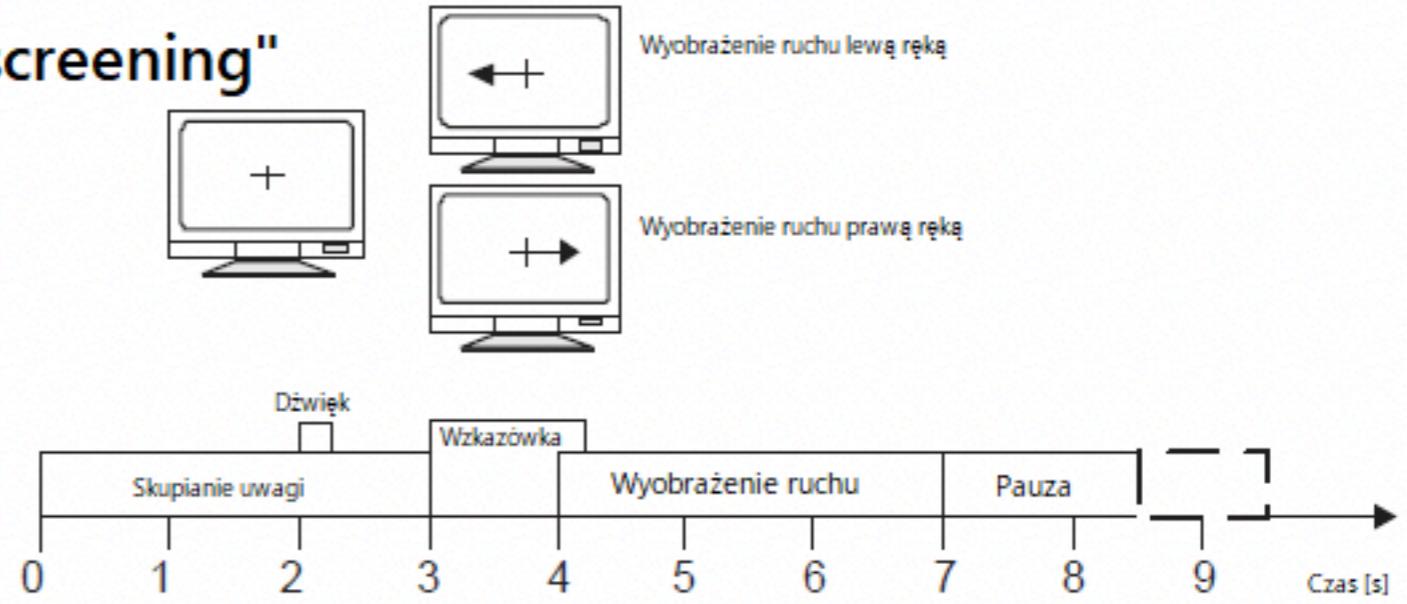
Przy samych wyobrażeniach analogicznych ruchów zmiany są podobne.

Kolory odpowiadają procentowej zmianie mocy względem poprzedzającego okresu 2s, zgodnie ze skalą barw umieszczoną po prawej stronie. Na poziomej osi - czas w s, na pionowej częstość w Hz.

# Przykładowe dane: BCI competition IV dataset 2b

(a) sesja "screening"

- wskazane wyobrażenia ruchowe ręka lewa albo prawa
- 3 kanały EEG: filtrowanie: 0.5-100Hz; filtr sieciowy,
- 250Hz próbkowanie,
- 2 klasy,
- 9 osób
- 6 serii po 20 prób na klasę epok danych na osobę, łącznie 2160 przykładów



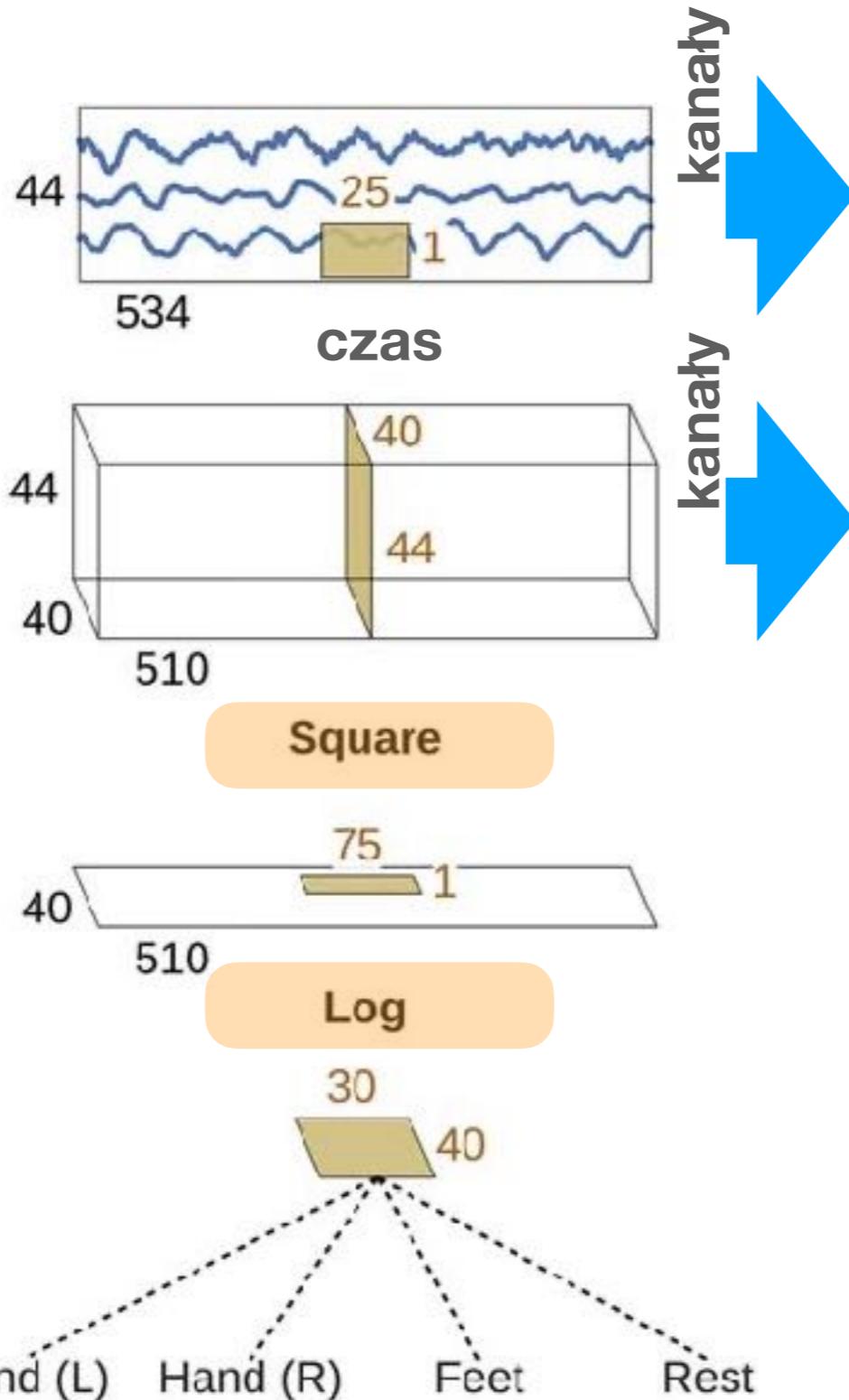
# Architektury

filtry  
czasowe  
40 Units

filtry  
przestrzenne  
40 Units

Mean Pooling  
Stride 15x1

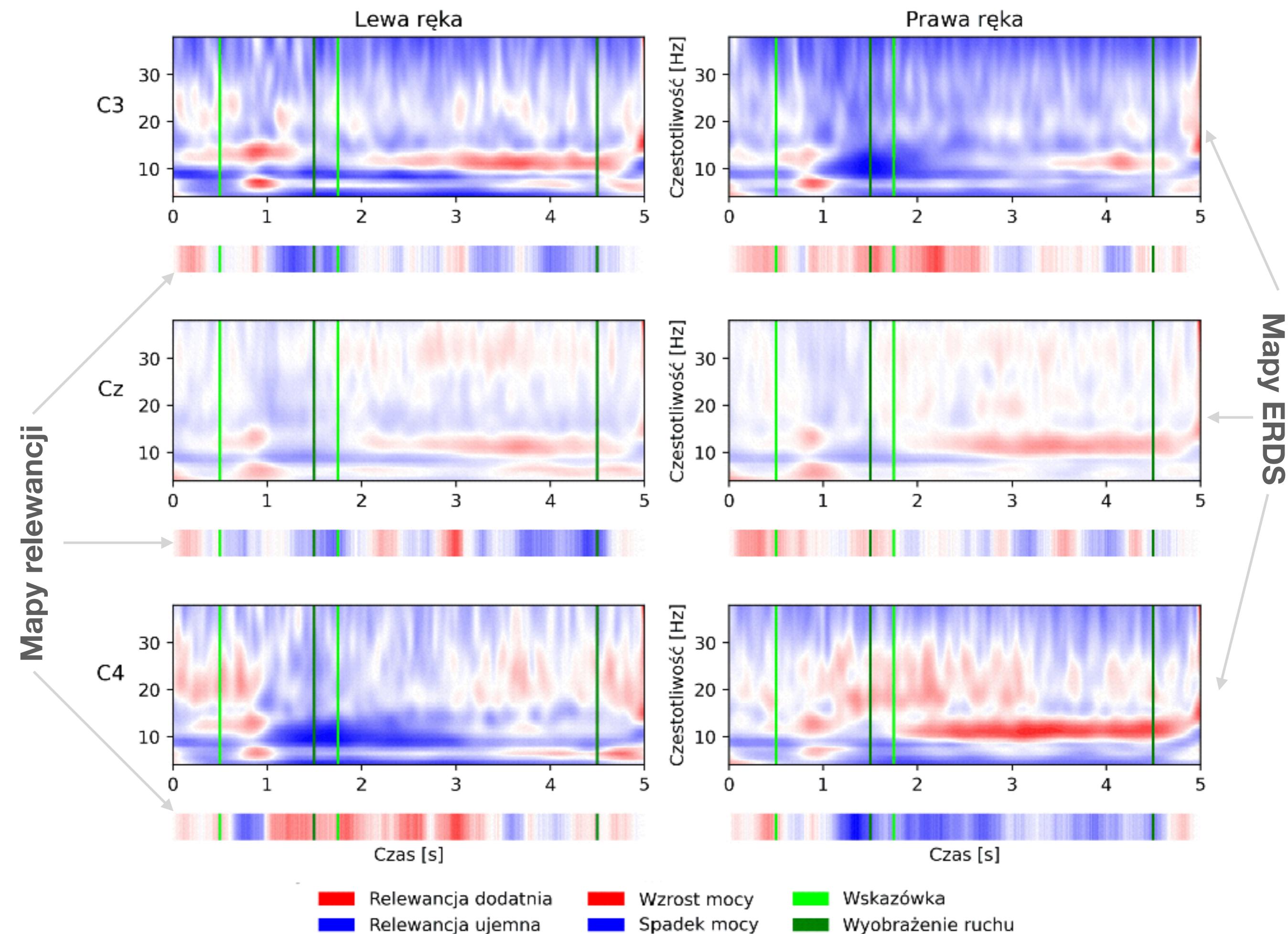
Linear Classification  
(Dense Layer+Softmax)  
4 Units



dostajemy 44 sygnały,  
każdy przefiltrowany czasowo  
na 40 sposobów

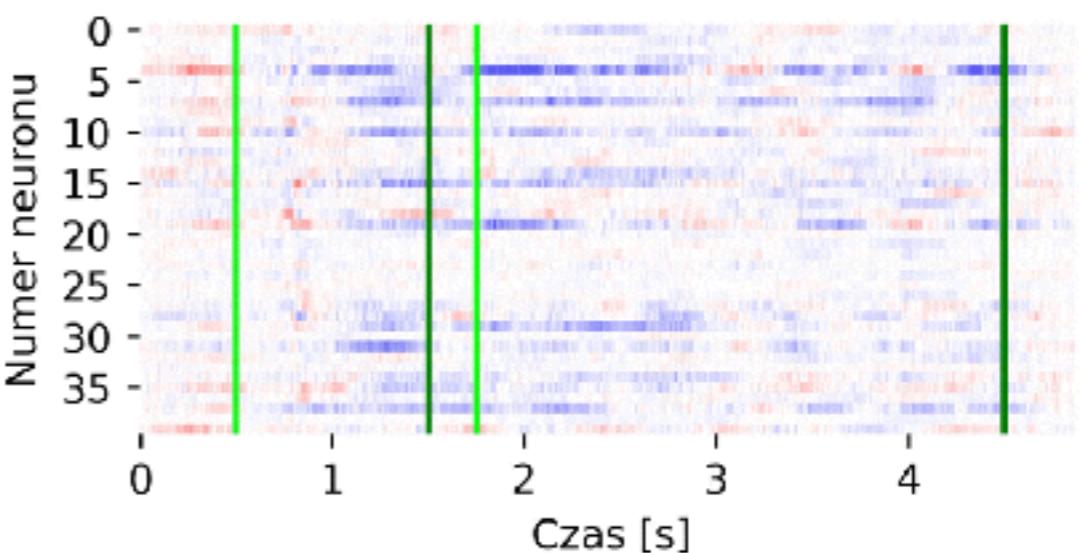
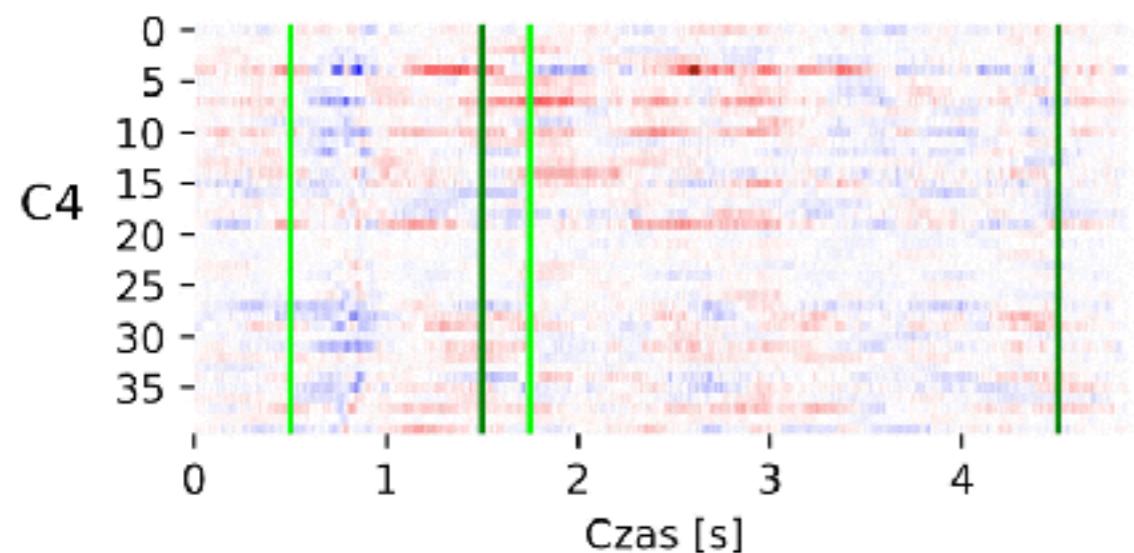
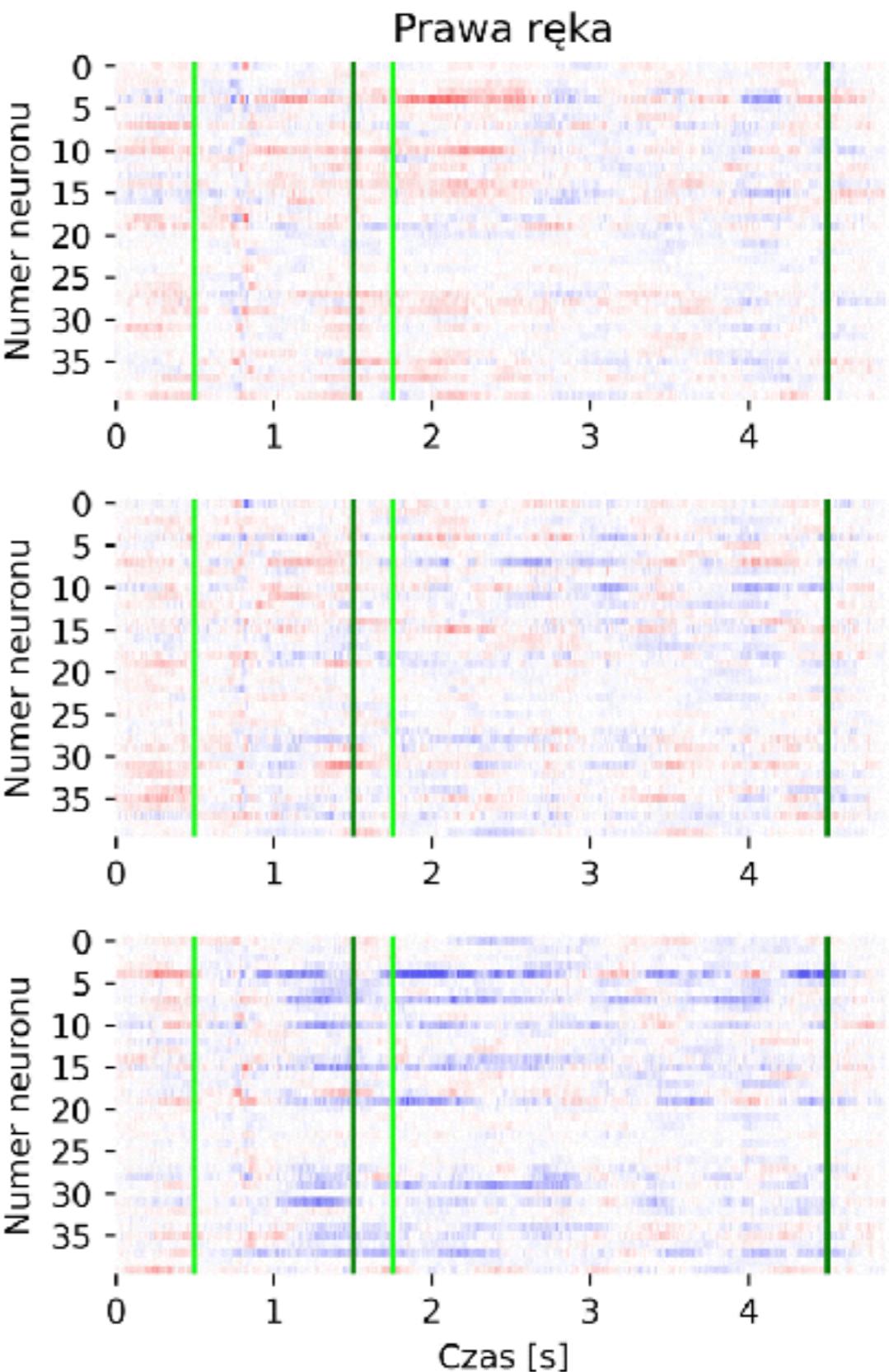
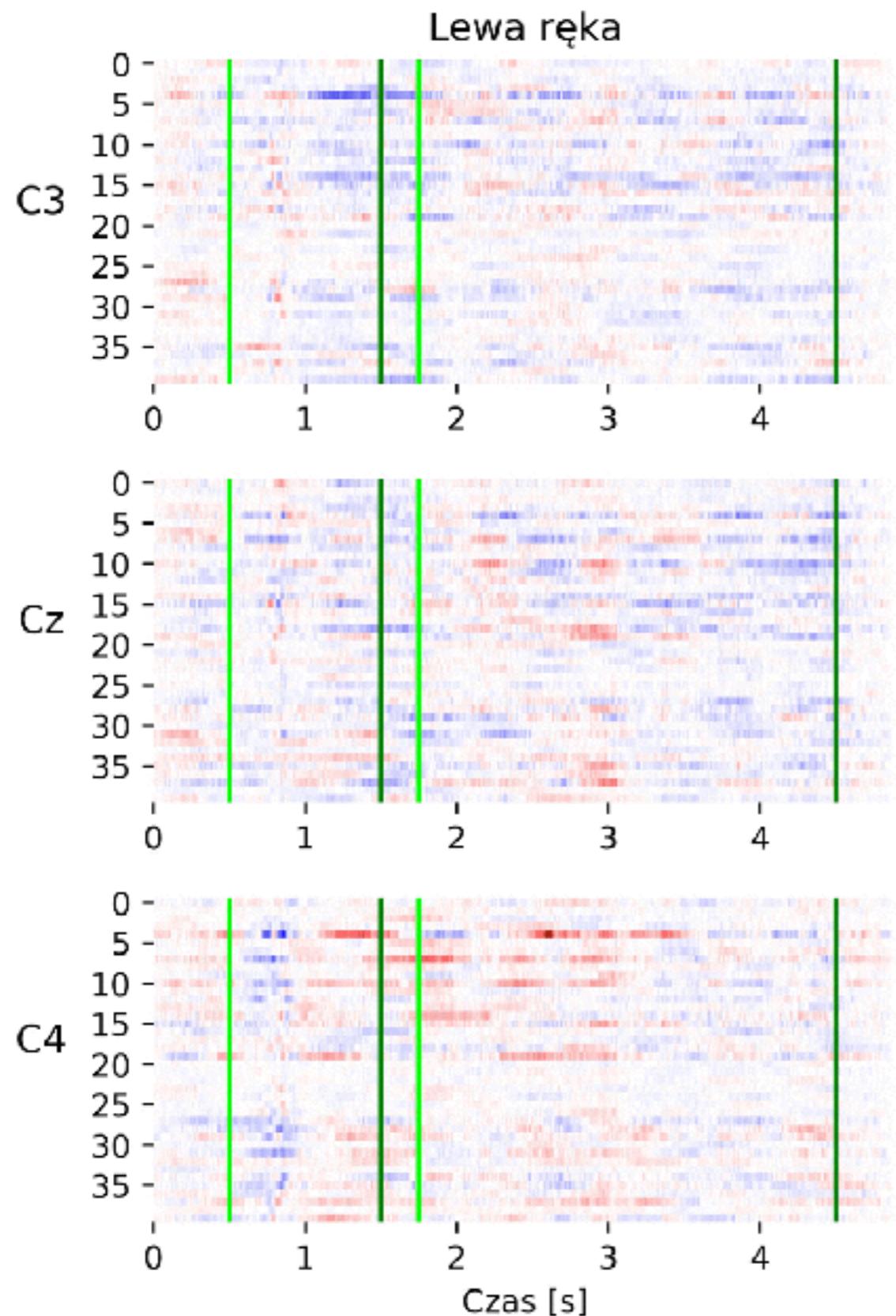
dostajemy 40 różnych  
liniowych kombinacji  
 $44 \text{ (syg.)} \times 40 \text{ (filt. czasowych)}$

**LRP dla warstwy  
wejściowej**  
**istotne momenty w czasie**



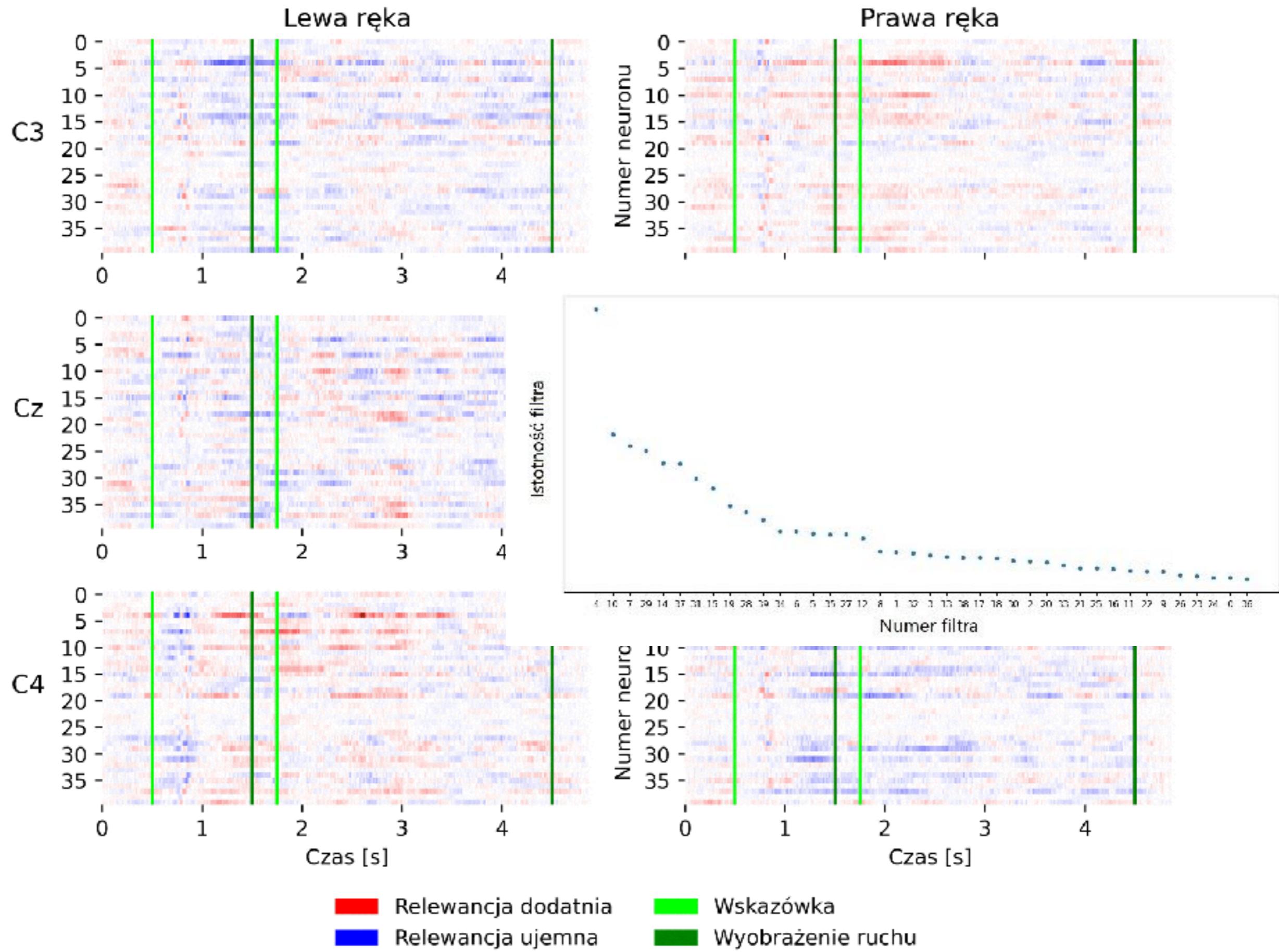
# **LRP dla pierwszej warstwy konwolucyjnej**

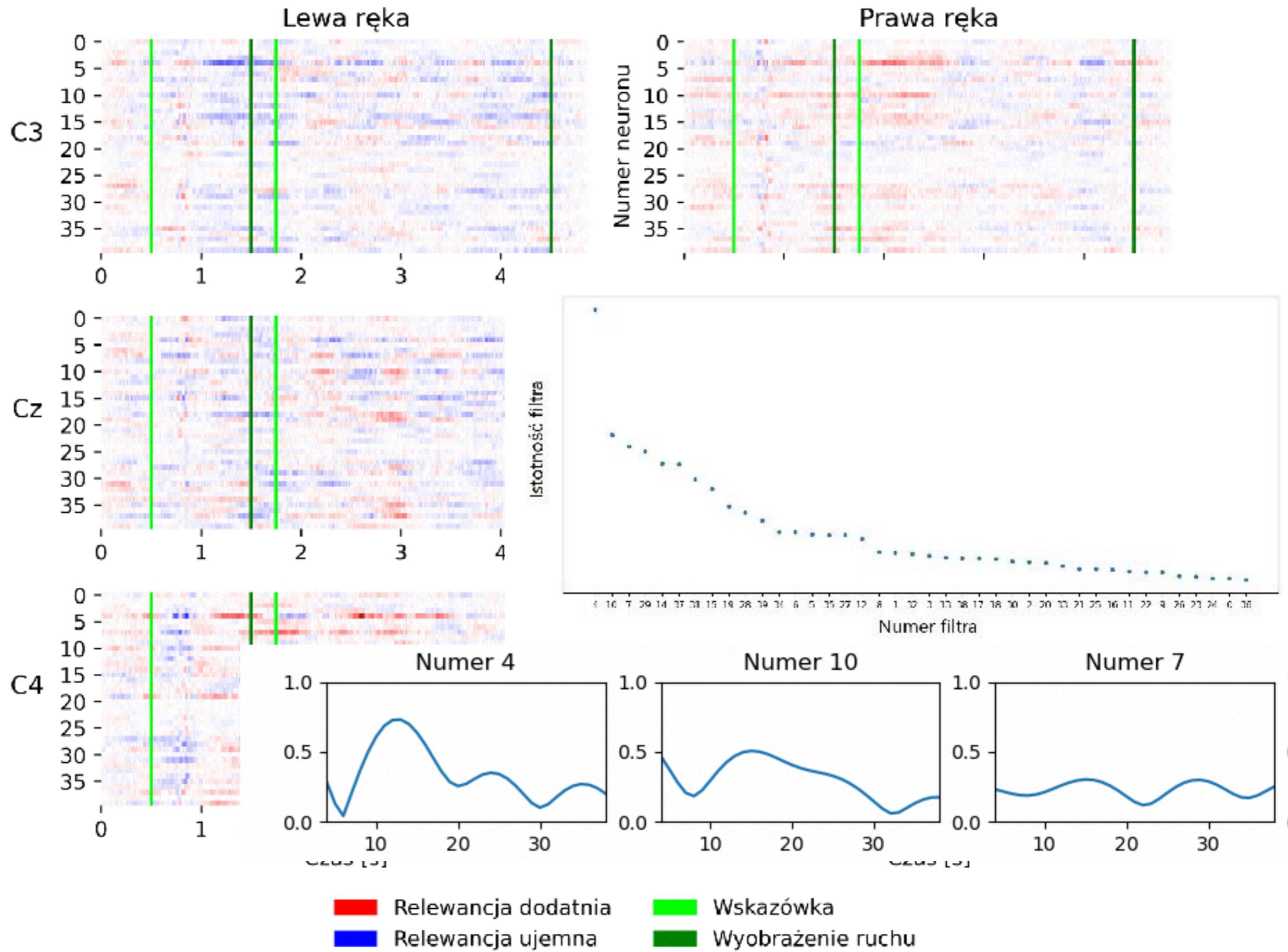
interpretujemy ja jako filtry  
częstotliwościowe FIR



■ Relewanca dodatnia  
■ Relewanca ujemna

■ Wskazówka  
■ Wyobrażenie ruchu





# więcej o technice LRP:

<http://www.heatmapping.org/>

toolbox w Keras z implementacją tej oraz innych technik

<https://github.com/albermax/investigate>

# **Sieć konwolucyjna z wbudowanym elementem wyjaśniającym CAM (class activation mapping)**

**Na podstawie:**

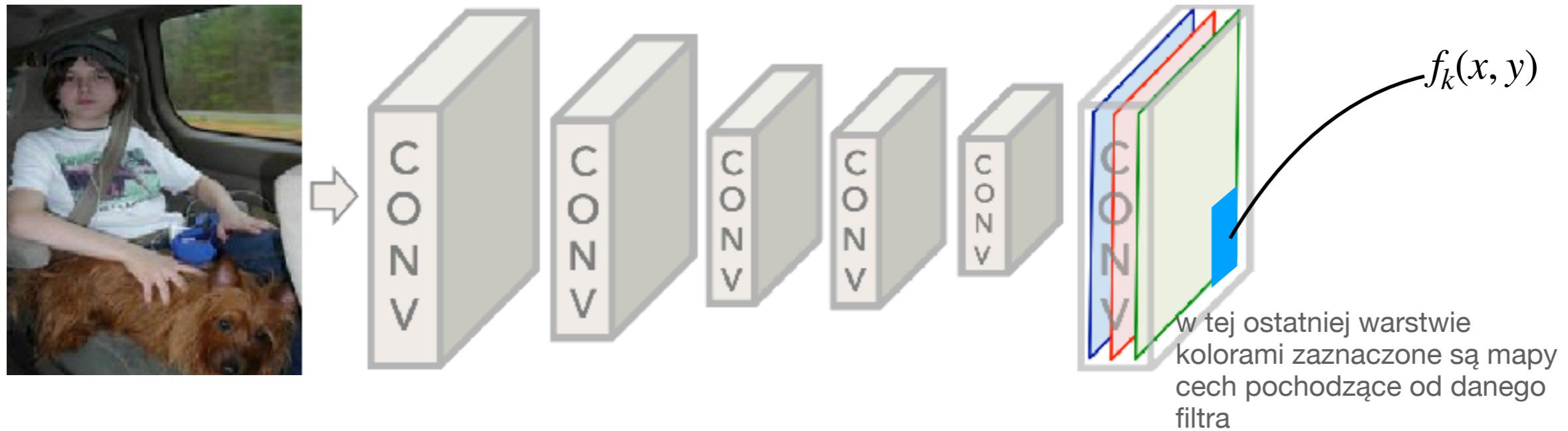
**Learning Deep Features for Discriminative Localization**

[Bolei Zhou](#), [Aditya Khosla](#), [Agata Lapedriza](#), [Aude Oliva](#), [Antonio Torralba](#)

<https://arxiv.org/abs/1512.04150v1>

# Sieć konwolucyjna z wbudowanym elementem wyjaśniającym CAM (class activation mapping)

- sieci konwolucyjne (bez warstw FC) zachowują informację przestrzenną w mapach aktywacji na kolejnych warstwach



Na podstawie:

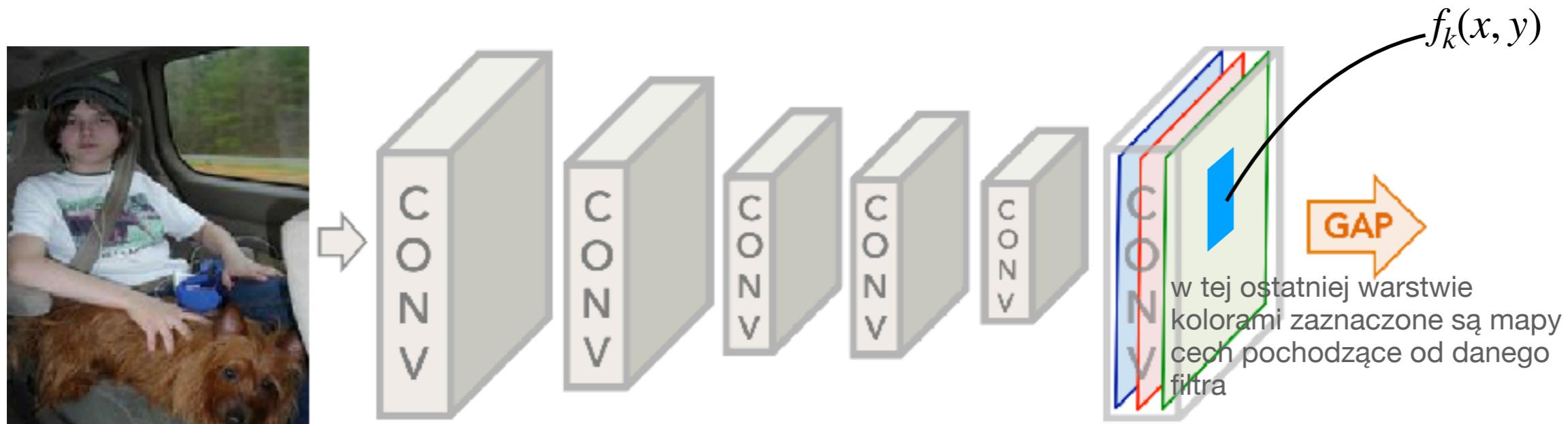
**Learning Deep Features for Discriminative Localization**

[Bolei Zhou](#), [Aditya Khosla](#), [Agata Lapedriza](#), [Aude Oliva](#), [Antonio Torralba](#)

<https://arxiv.org/abs/1512.04150v1>

# Sieć konwolucyjna z wbudowanym elementem wyjaśniającym CAM (class activation mapping)

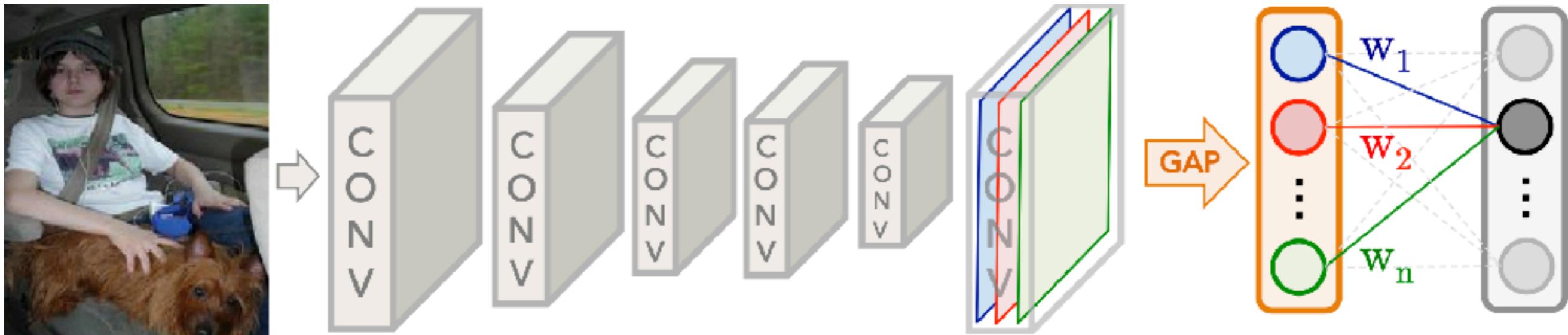
- sieci konwolucyjne (bez warstw FC) zachowują informację przestrzenną w mapach aktywacji na kolejnych warstwach



Global Average Pooling (GAP):  $F_k = \sum_{x,y} f_k(x, y)$  - sumaryczna aktywacja k-tego filtra dla danego wejścia

# Sieć konwolucyjna z wbudowanym elementem wyjaśniającym CAM (class activation mapping)

- sieci konwolucyjne (bez warstw FC) zachowują informację przestrzenną w mapach aktywacji na kolejnych warstwach



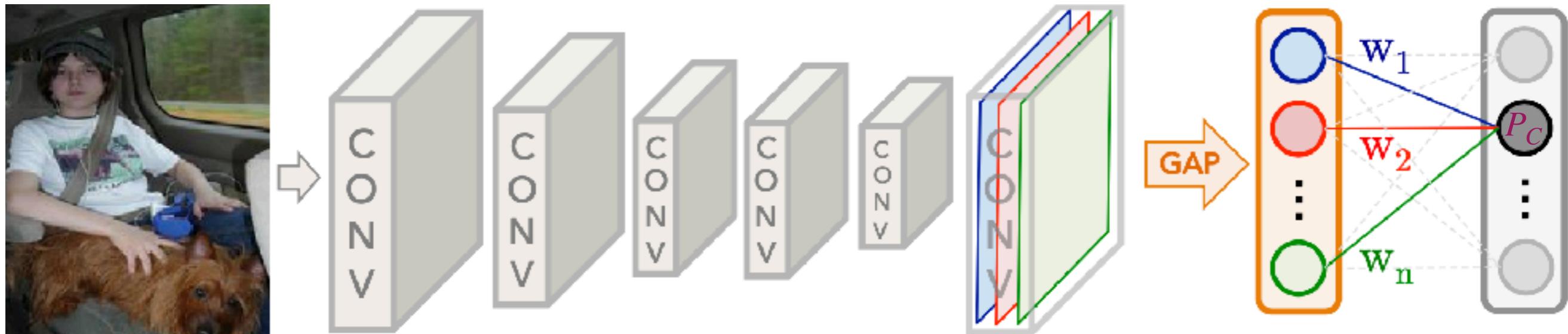
Global Average Pooling (GAP):  $F_k = \sum_{x,y} f_k(x, y)$  - sumaryczna aktywacja k-tego filtra dla danego wejścia

Na wyjściu dołożymy Softmax. Jego wejście dla klasy C to:

$$S_C = \sum_k w_k^C F_k$$

# Sieć konwolucyjna z wbudowanym elementem wyjaśniającym CAM (class activation mapping)

- sieci konwolucyjne (bez warstw FC) zachowują informację przestrzenną w mapach aktywacji na kolejnych warstwach



Global Average Pooling (GAP):  $F_k = \sum_{x,y} f_k(x, y)$  - sumaryczna aktywacja k-tego filtra dla danego wejścia

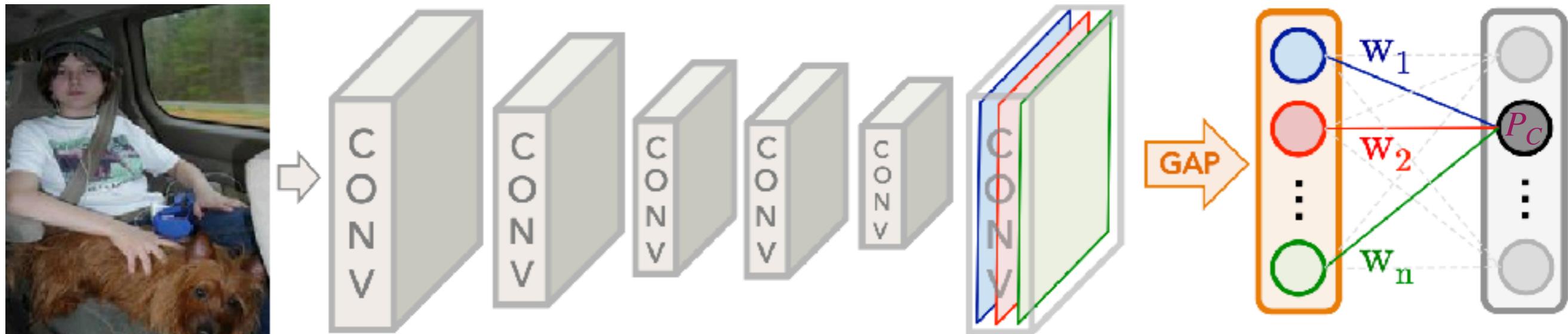
Na wyjściu dołożymy Softmax. Jego wejście dla klasy C to:

$$S_C = \sum_k w_k^C F_k$$

Zaś wyjście dla klasy C to jej prawdopodobieństwo:  $P_C = \frac{\exp S_C}{\sum_K S_K}$

# Sieć konwolucyjna z wbudowanym elementem wyjaśniającym CAM (class activation mapping)

- sieci konwolucyjne (bez warstw FC) zachowują informację przestrzenną w mapach aktywacji na kolejnych warstwach



Global Average Pooling (GAP):  $F_k = \sum_{x,y} f_k(x,y)$  - sumaryczna aktywacja k-tego filtra dla danego wejścia

Na wyjściu dołożymy Softmax. Jego wejście dla klasy C to:

$$S_C = \sum_k w_k^C F_k = \sum_k w_k^C \sum_{x,y} f_k(x,y) = \sum_k \sum_{x,y} w_k^C f_k(x,y) = \sum_{x,y} \underbrace{\sum_k w_k^C f_k(x,y)}_{M_C(x,y)} = \sum_{x,y} M_C(x,y)$$

To pozwala interpretować  $M_C(x,y)$  jako mapę istotności cech dla klasy C w pozycji (x,y)

Zaś wyjście dla klasy C to jej prawdopodobieństwo:  $P_C = \frac{\exp S_C}{\sum_K S_K}$

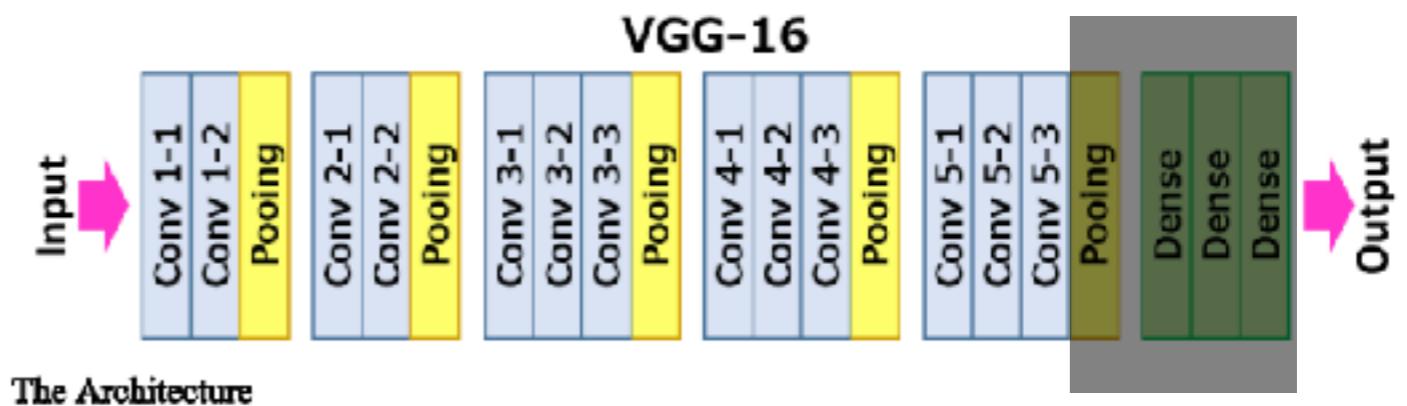
# Sieć konwolucyjna z wbudowanym elementem wyjaśniającym CAM (class activation mapping)

- sieci konwolucyjne (bez warstw FC) zachowują informację przestrzenną w mapach aktywacji na kolejnych warstwach

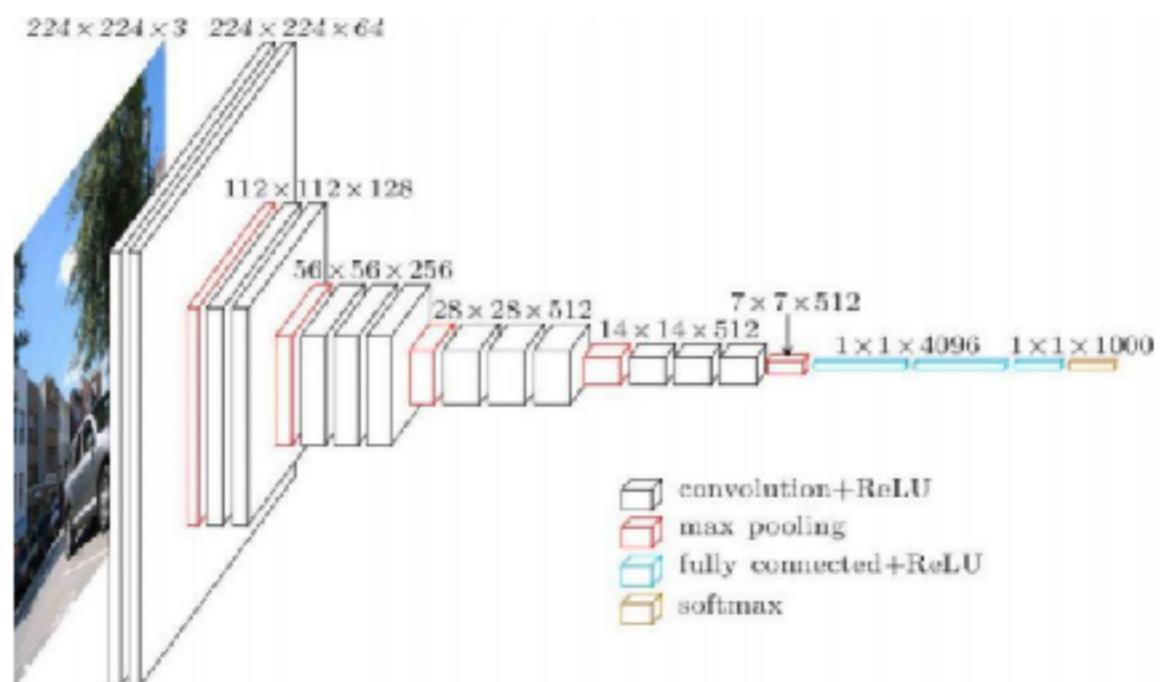


# Jak uczyć CAM?

- bierzemy przetrenowaną sieć np. VGGNet
- obcinamy warstwy w pełni połączone (Dense)
- dokładamy GAP i softmax
- trenujemy do naszego problemu



The architecture depicted below is VGG16.



Uwaga autorów:

„Note that it is important for the networks to perform well on classification in order to achieve a high performance on localization as it involves identifying both the object category and the bounding box location accurately. „

# Rozumienie modelu

- chodzi tu o rozumienie funkcjonalne, a nie o rozumienie algorytmiki modelu
- rozumienie zachowania vs. rozumienie obliczeń wykonywanych wewnątrz czarnej skrzynki

