Wykorzystanie sieci neuronowych w optymalizacji grafiki i animacji

Dr inż. Marek Woda

Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Rozwoju Regionalnego Program Operacyjny Polska Cyfrowa na lata 2014-2020,
Oś Priorytetowa nr 3 "Cyfrowe kompetencje społeczeństwa" Działanie nr 3.2 "Innowacyjne rozwiązania na rzecz aktywizacji cyfrowej"
Tytuł projektu: „Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI Tech)”
Porozumienie nr POPC.03.02.00-00-0001/20-00

Plan wykładu

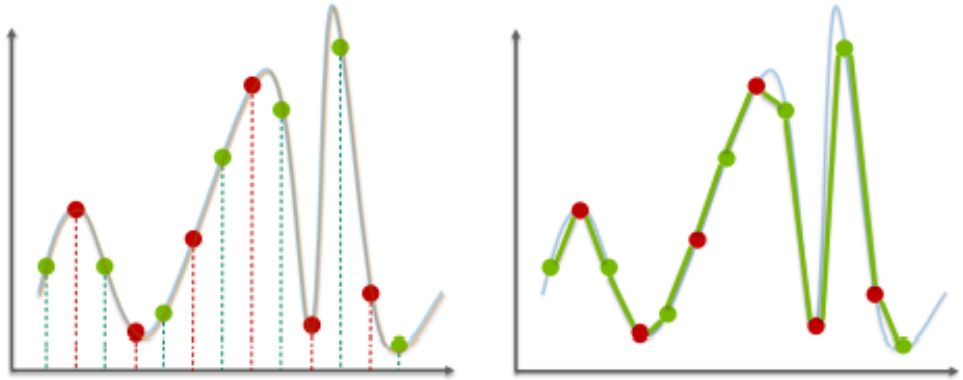
- Zjawisko aliasingu
- Techniki minimalizowania aliasingu (DLSS)
- Generatywne Sieci Adwersaryjne
- Definicje
- Model dyskryminatora i generatora
- Uczenie GANów –
- Najlepsze praktyki (DGAN)

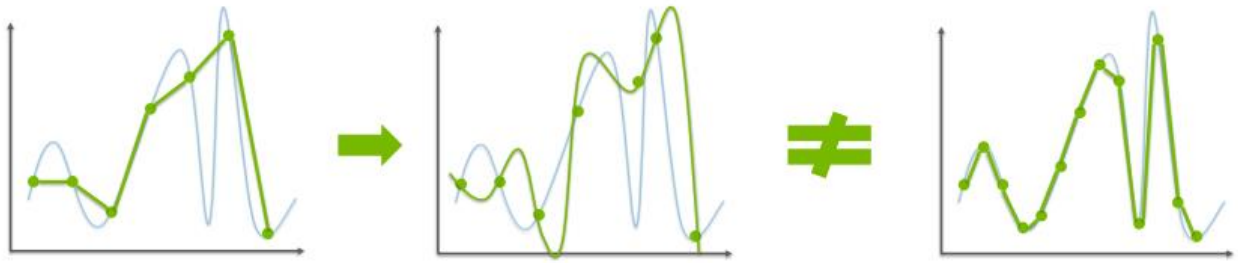
Aliasing

- wynik zmian w rysowanym obiekcie, które zachodzą szybciej niż odstępy między pikselami
- forma zaniżania próbkowania

Powód

- częstotliwość próbkowania zbyt mała, przebieg jest skonstruowany wygląda jak przebieg o niższej częstotliwości lub jak płaska linia





Aliasing

- w obrazach 2D objawia się jako:
  - wzór mory
  - rozpikselowane krawędzie, potocznie nazywane schodkami (ang. jaggies)

Zrzut ekranu z gry Battlefield 4 firmy EA

Aliasing

piksele

- rozmiar - ten sam
- kolor - jeden (z palety)

linia

- zbiór pikseli
- wydaje się „schodkowana”, chyba że jest idealnie pozioma lub pionowa

Aliasing

przetwarzanie sygnałów i obrazów

- doskonała eliminacja *aliasingu* ➔ próbkowanie przestrzenne z szybkością ***Nyquista*** (lub wyższą) po zastosowaniu filtra anty-aliasingowego 2D
- wymaga przeprowadzenia prostej i odwrotnej transformacji *Fouriera**

mniej wymagające obliczeniowo przybliżenia

- multi / supersampling pozwalają uniknąć przełączania domen ➔ pozostanie w domenie przestrzennej* ("domenie obrazu")

Aliasing

** Konwersja domeny przestrzenną w domenę częstotliwości*

- Rozważmy dwuetapowe przekształcenie nieposortowanej listy 12 liczb

(2, 3, 1, 2, 2, 0, 1, 1, 0, 1, 0, 0)

Etap 1: sortowanie listy po wartościach

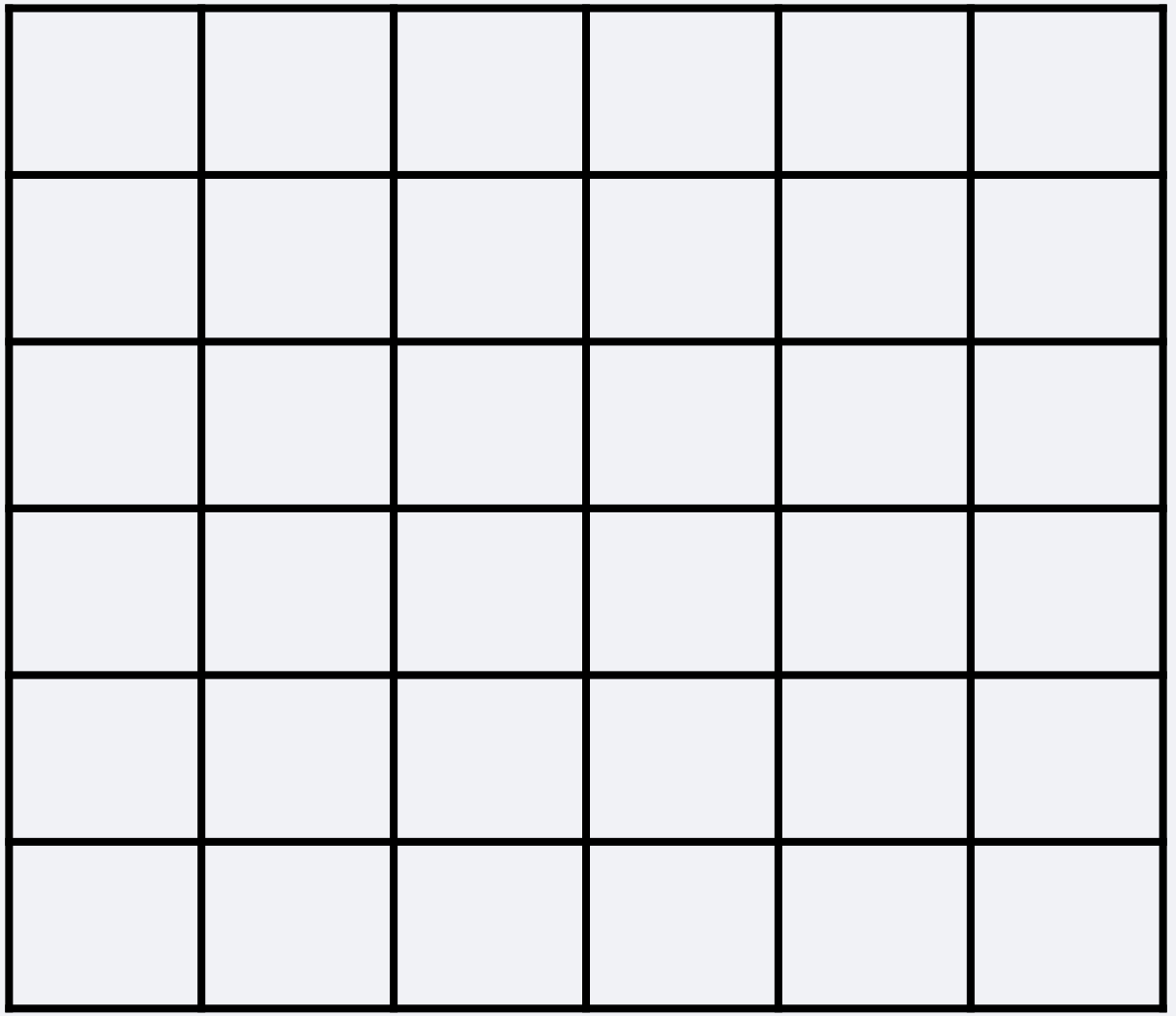
(0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 3,)

Etap 2: określenie występowania warości liczb

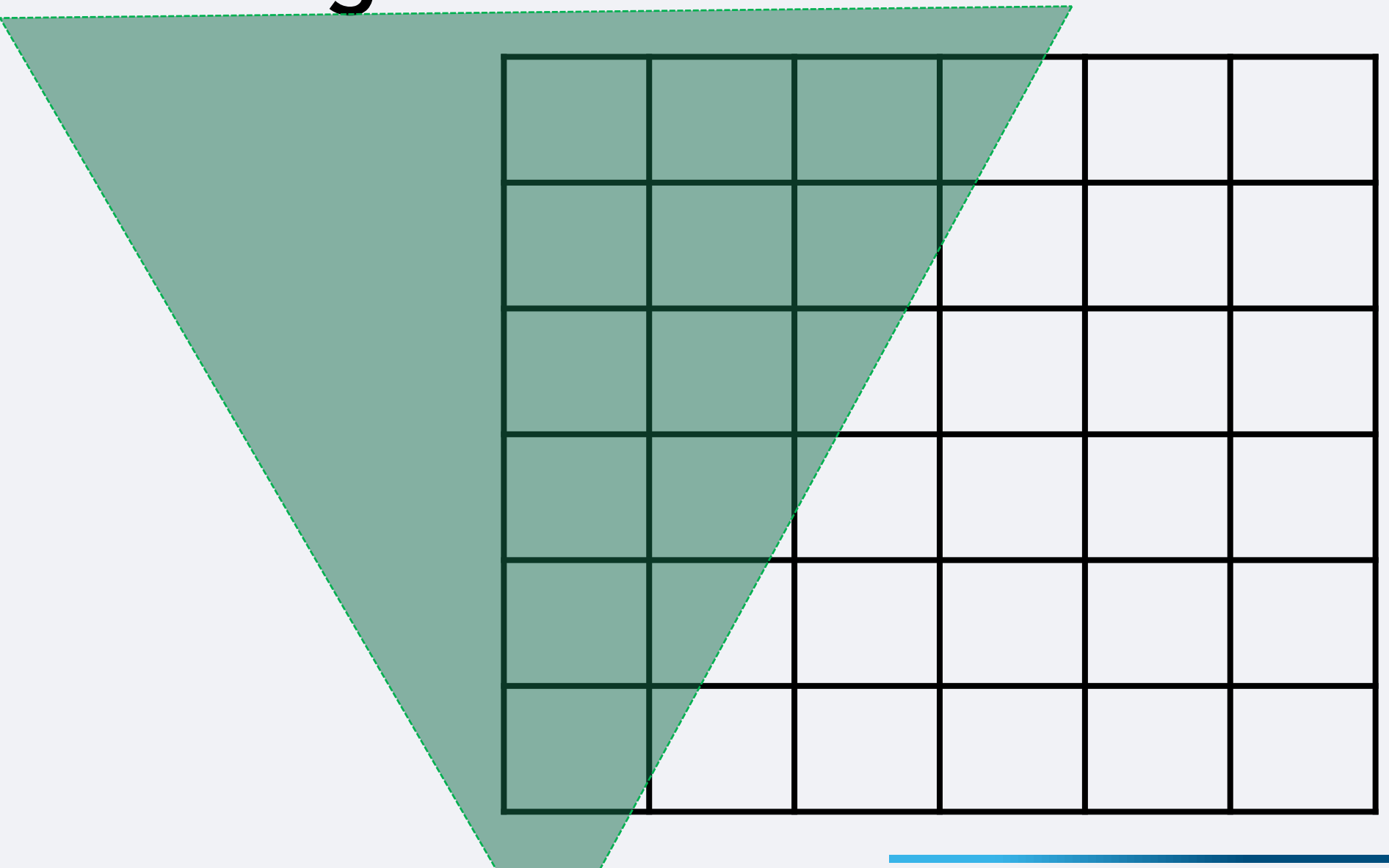
(4,4,3,1)

Wynik: *tracimy informację przestrzenną, zyskujemy informację o częstotliwości*

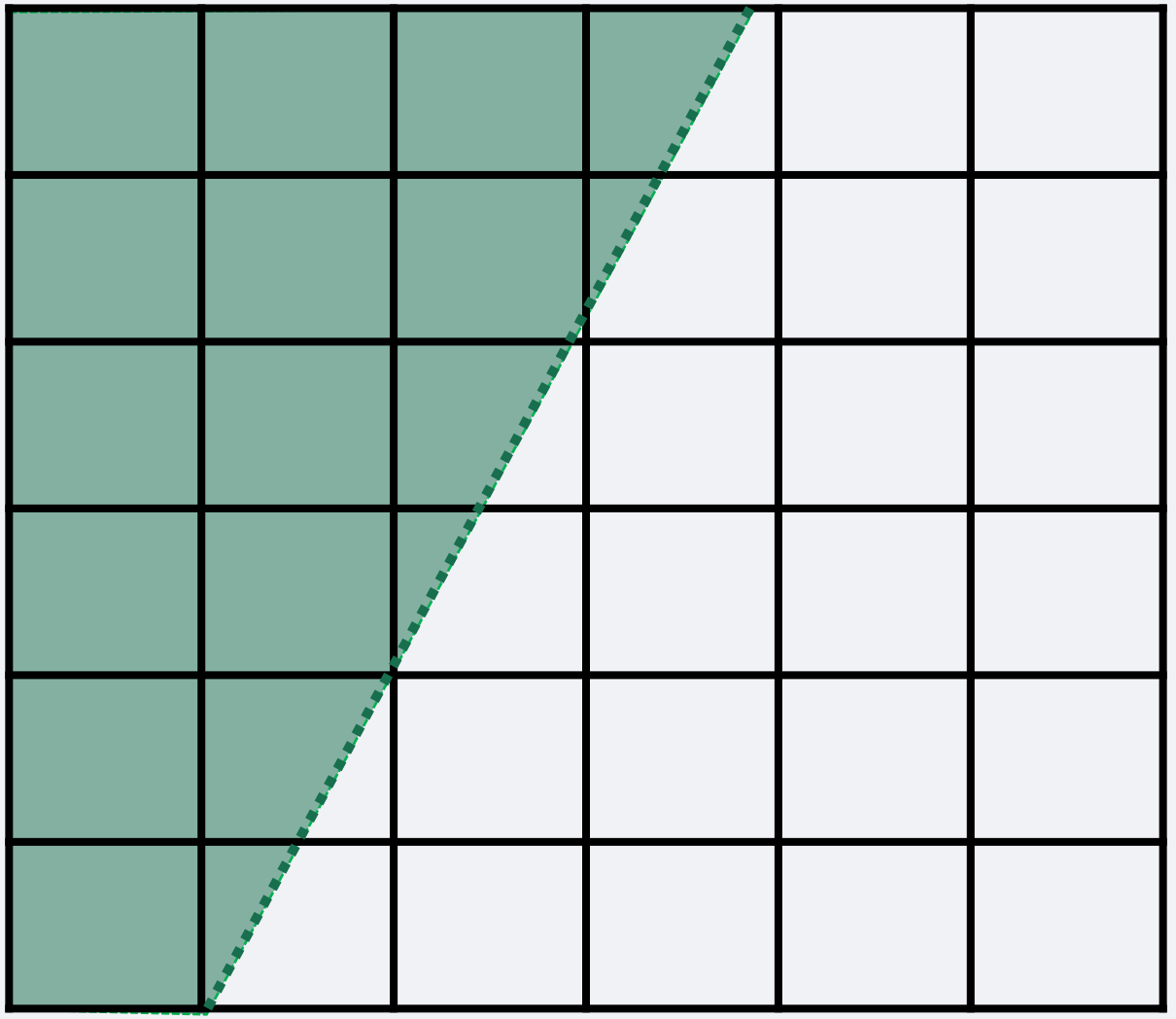
Aliasing



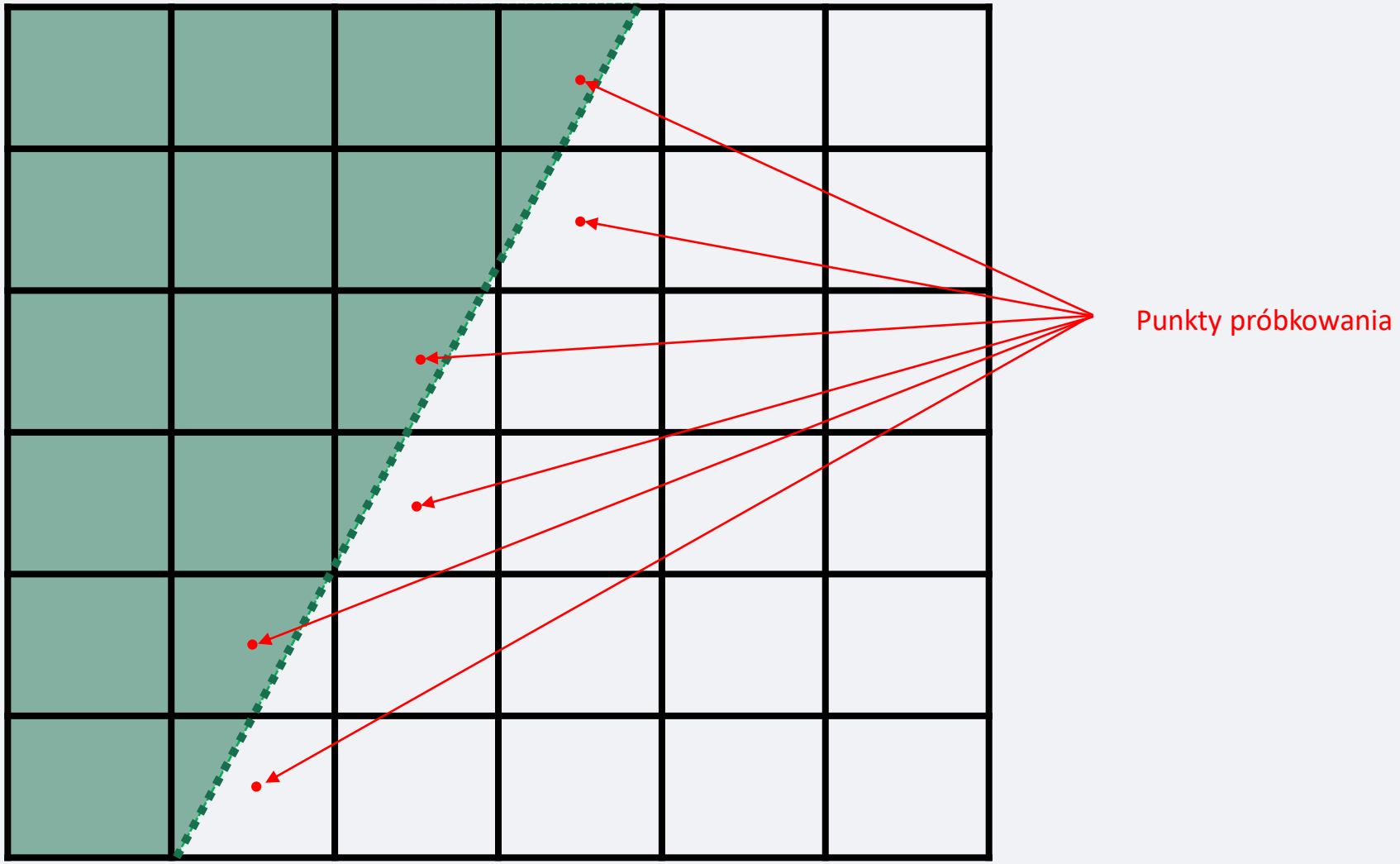
Aliasing



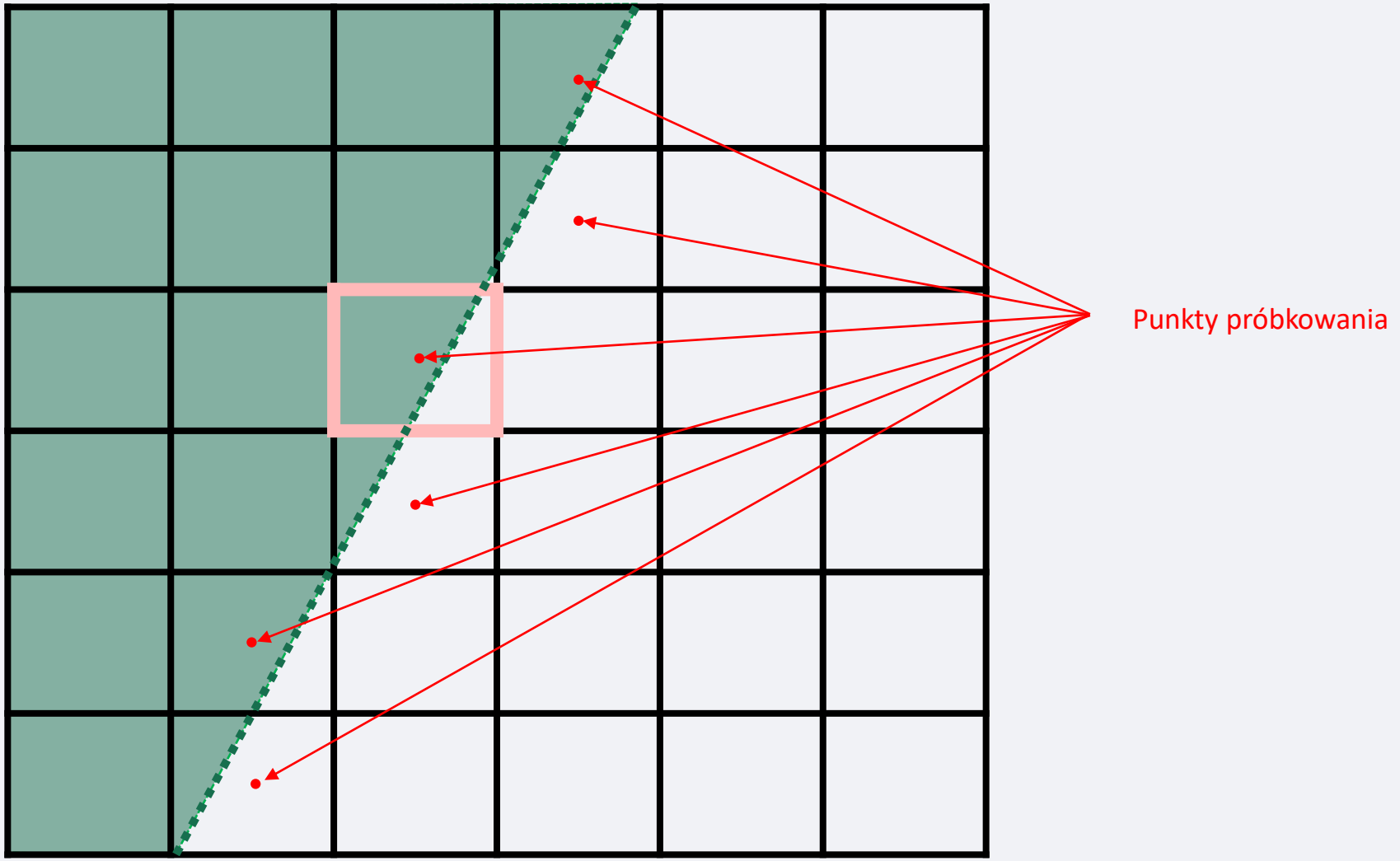
Aliasing



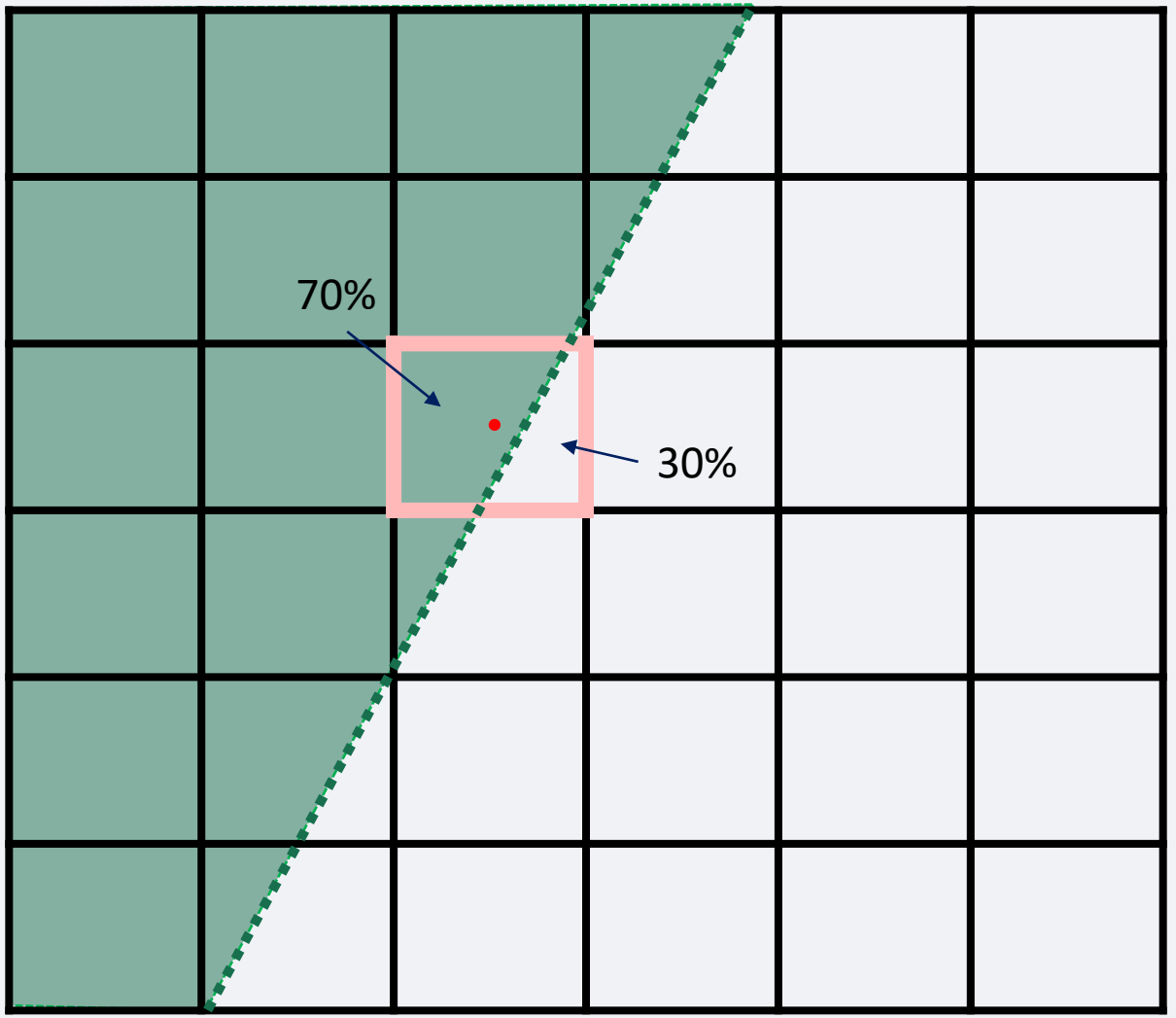
Aliasing



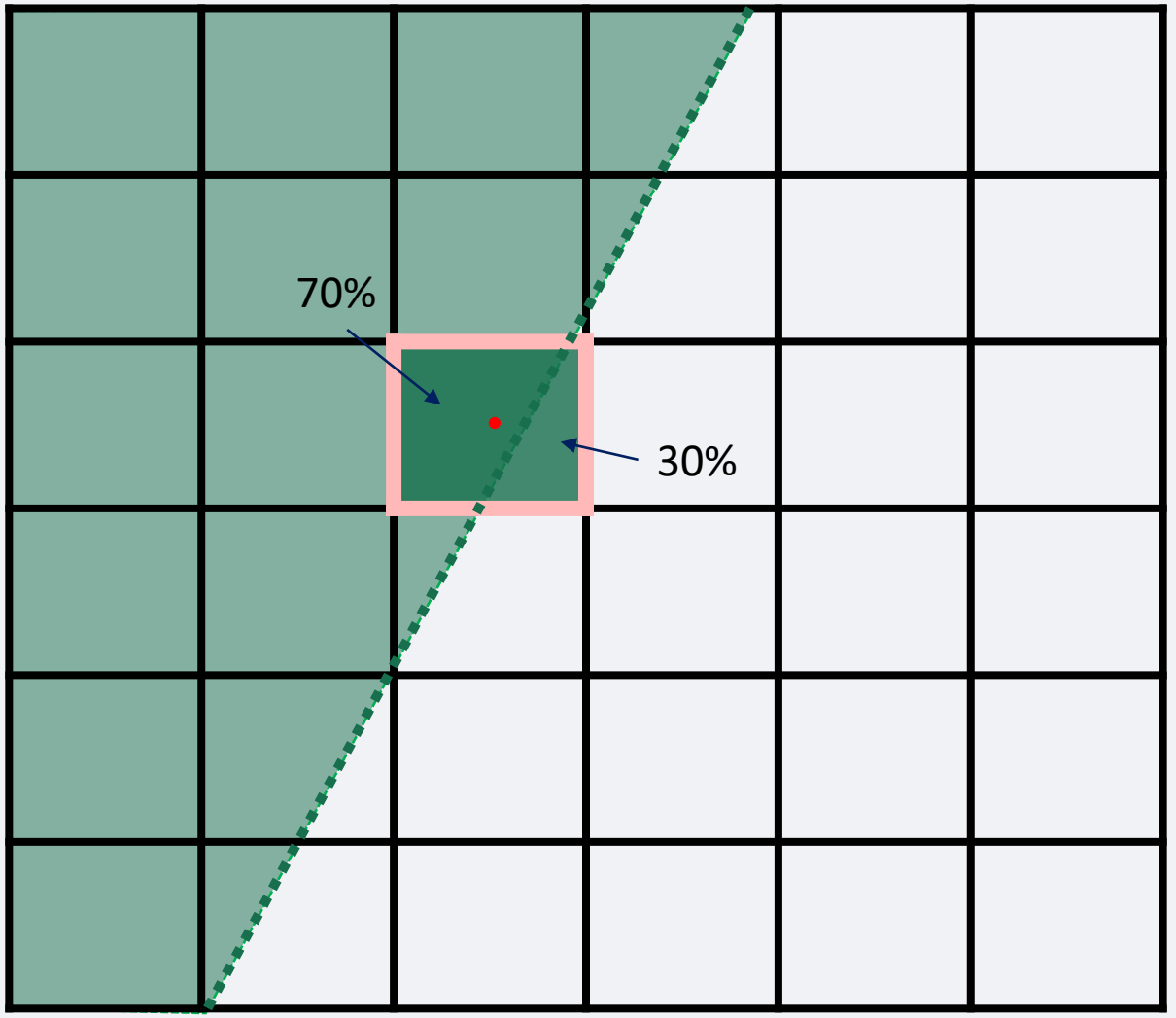
Aliasing



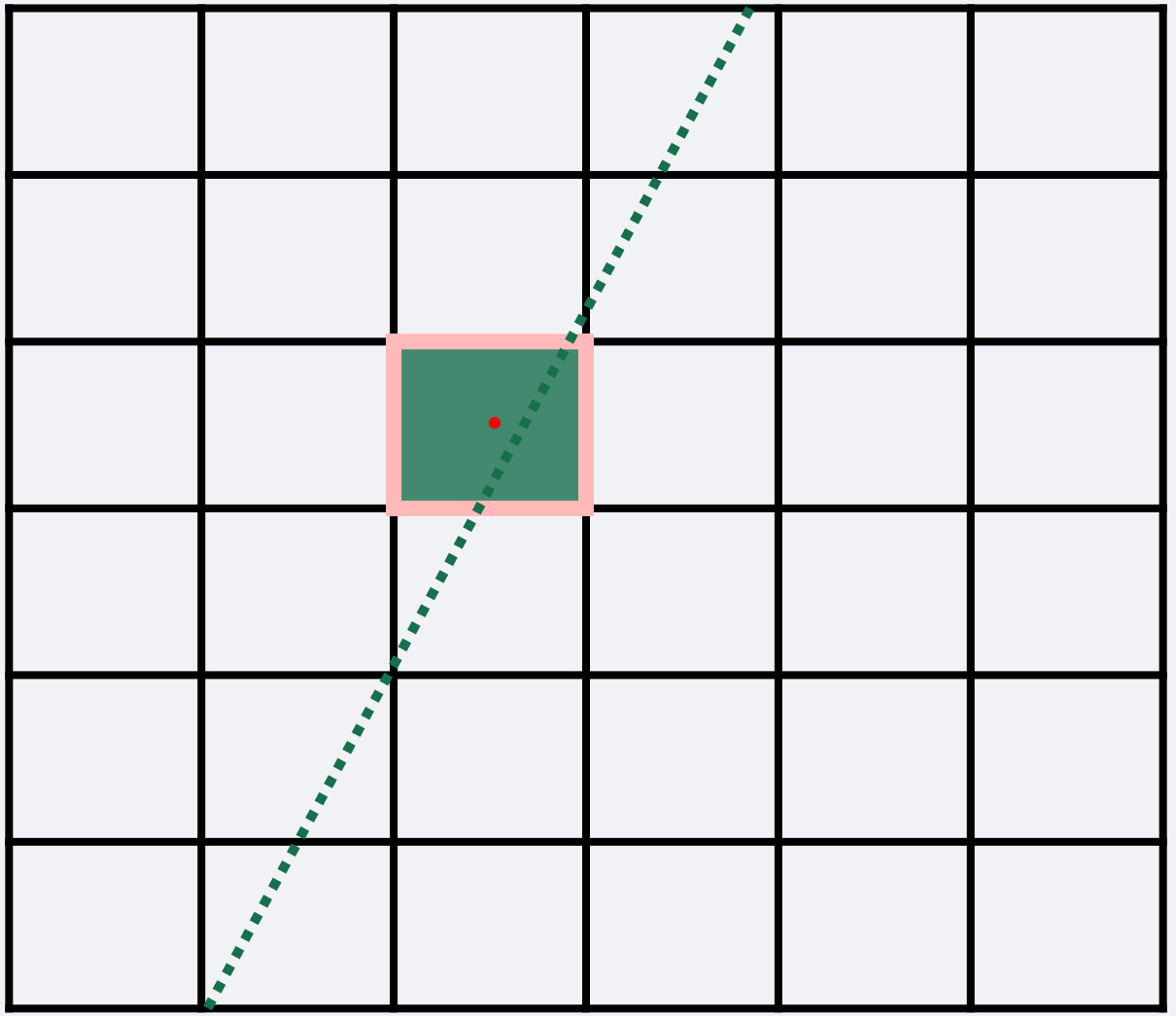
Aliasing



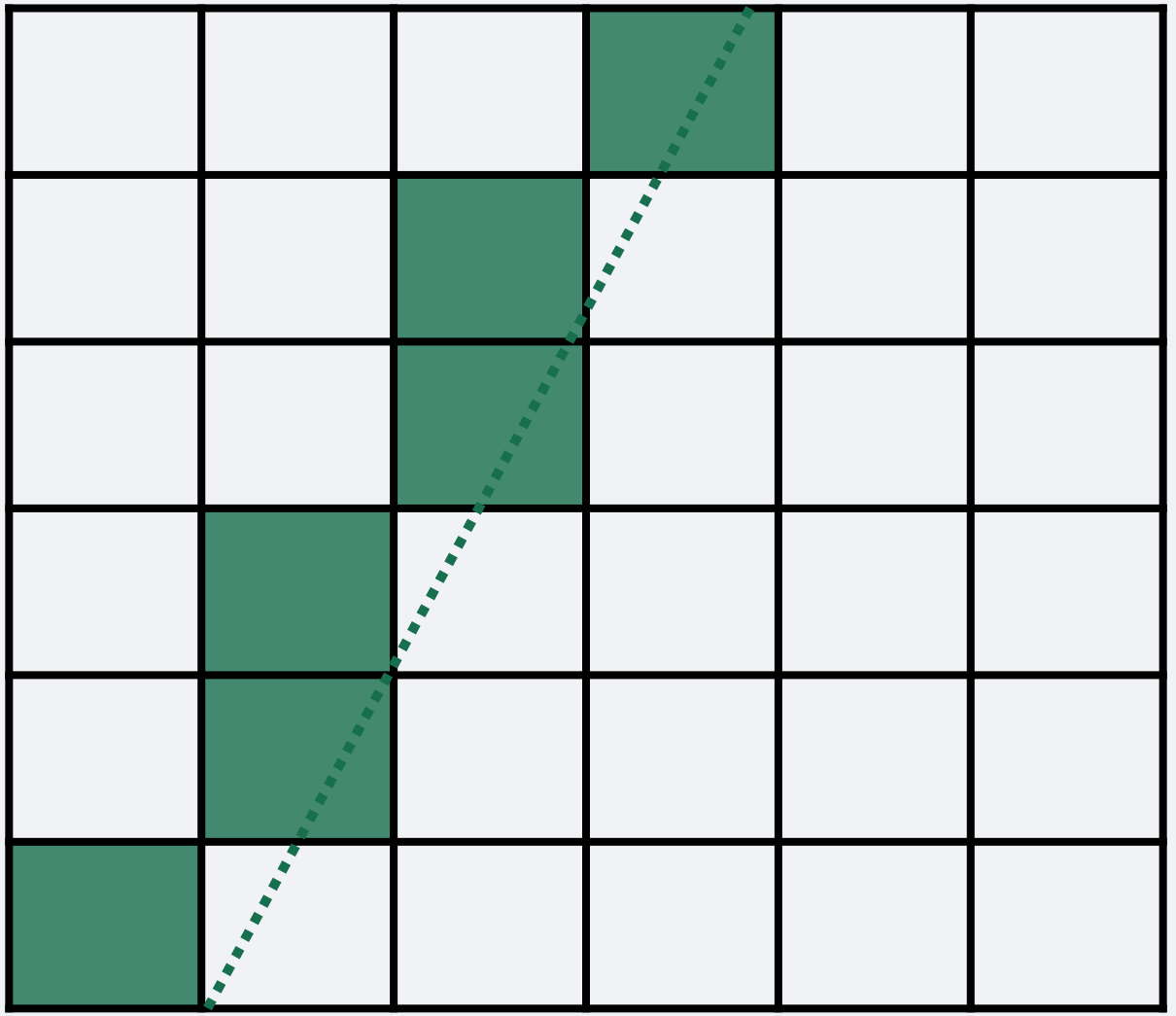
Aliasing



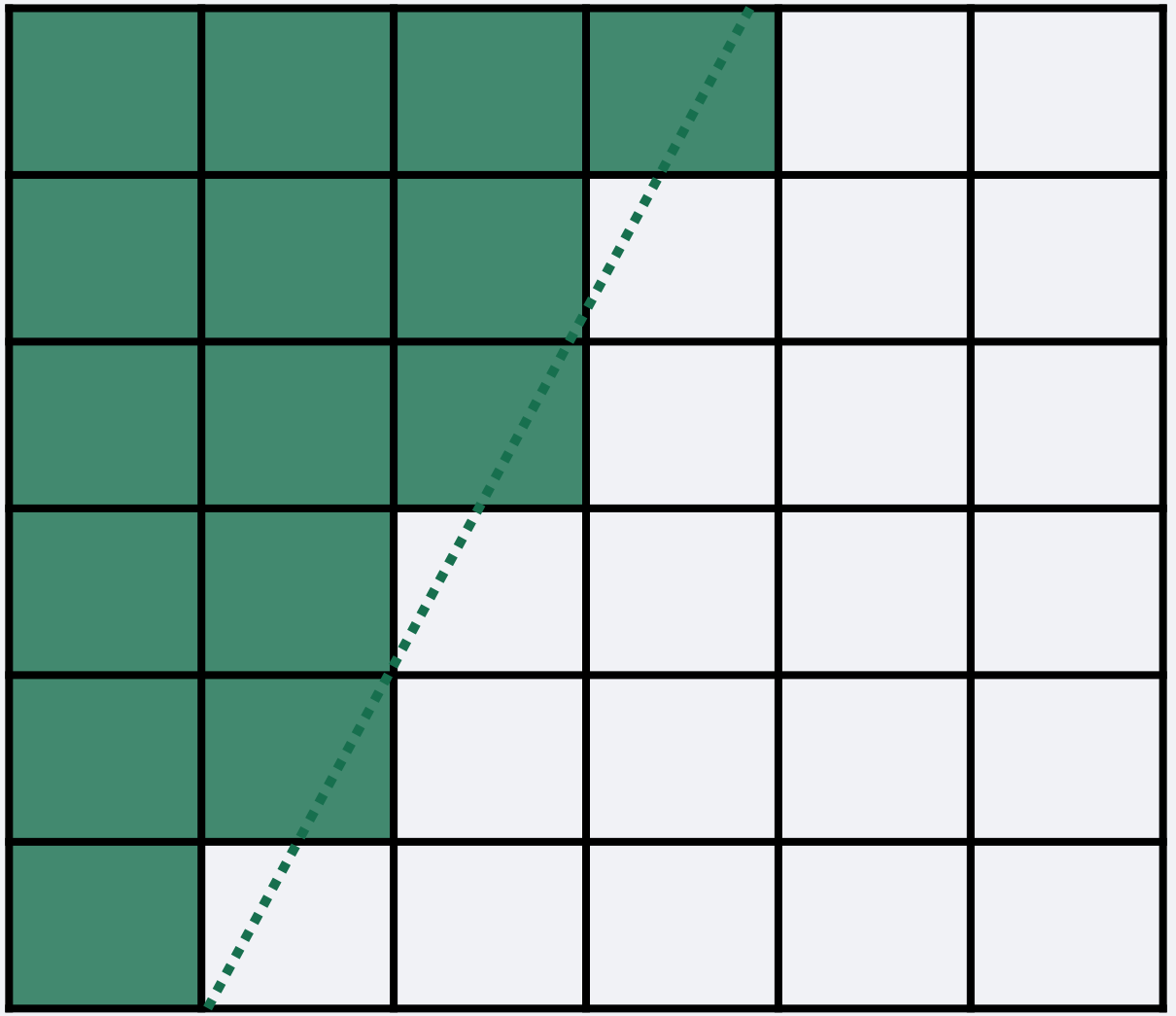
Aliasing



Aliasing



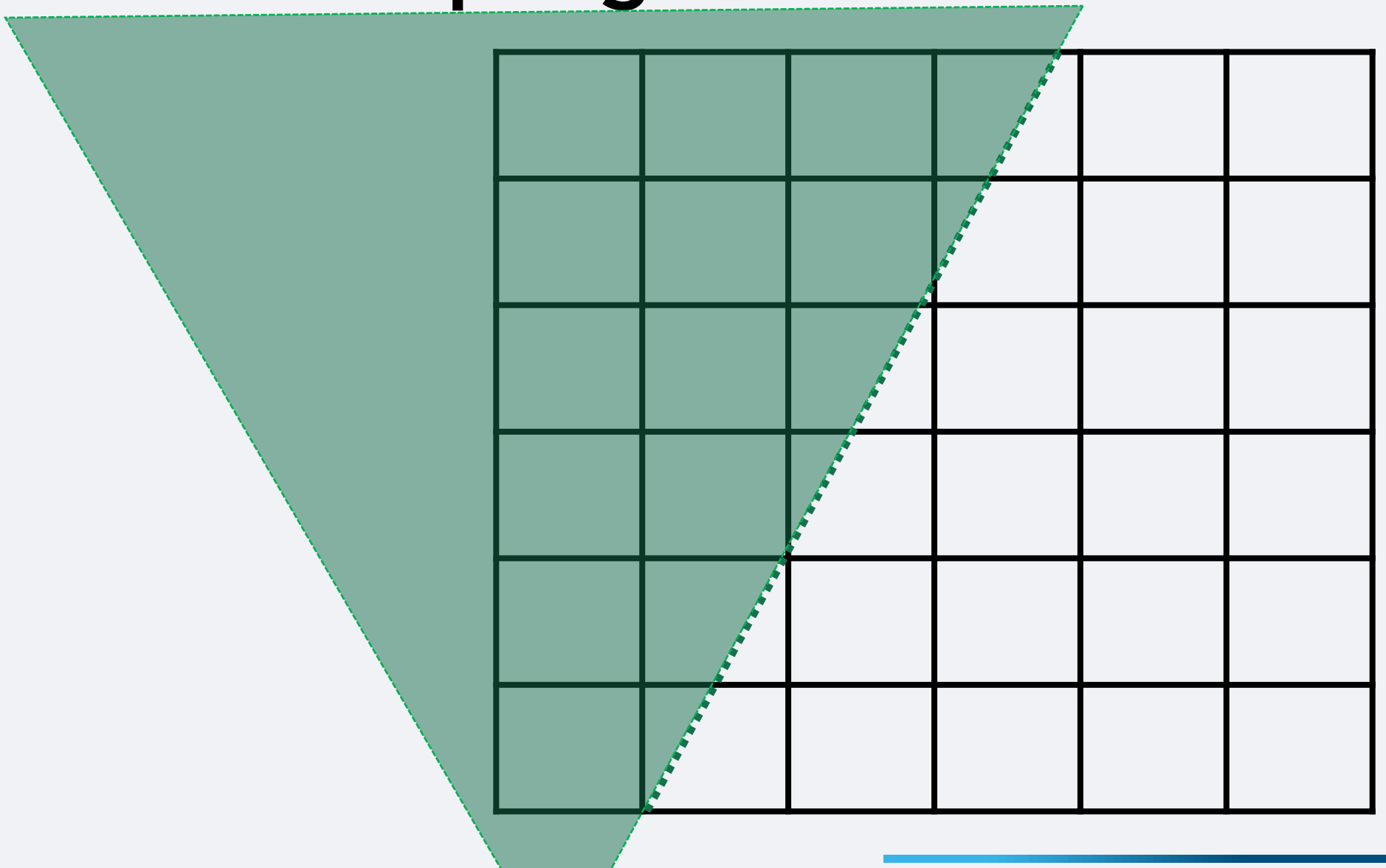
Aliasing



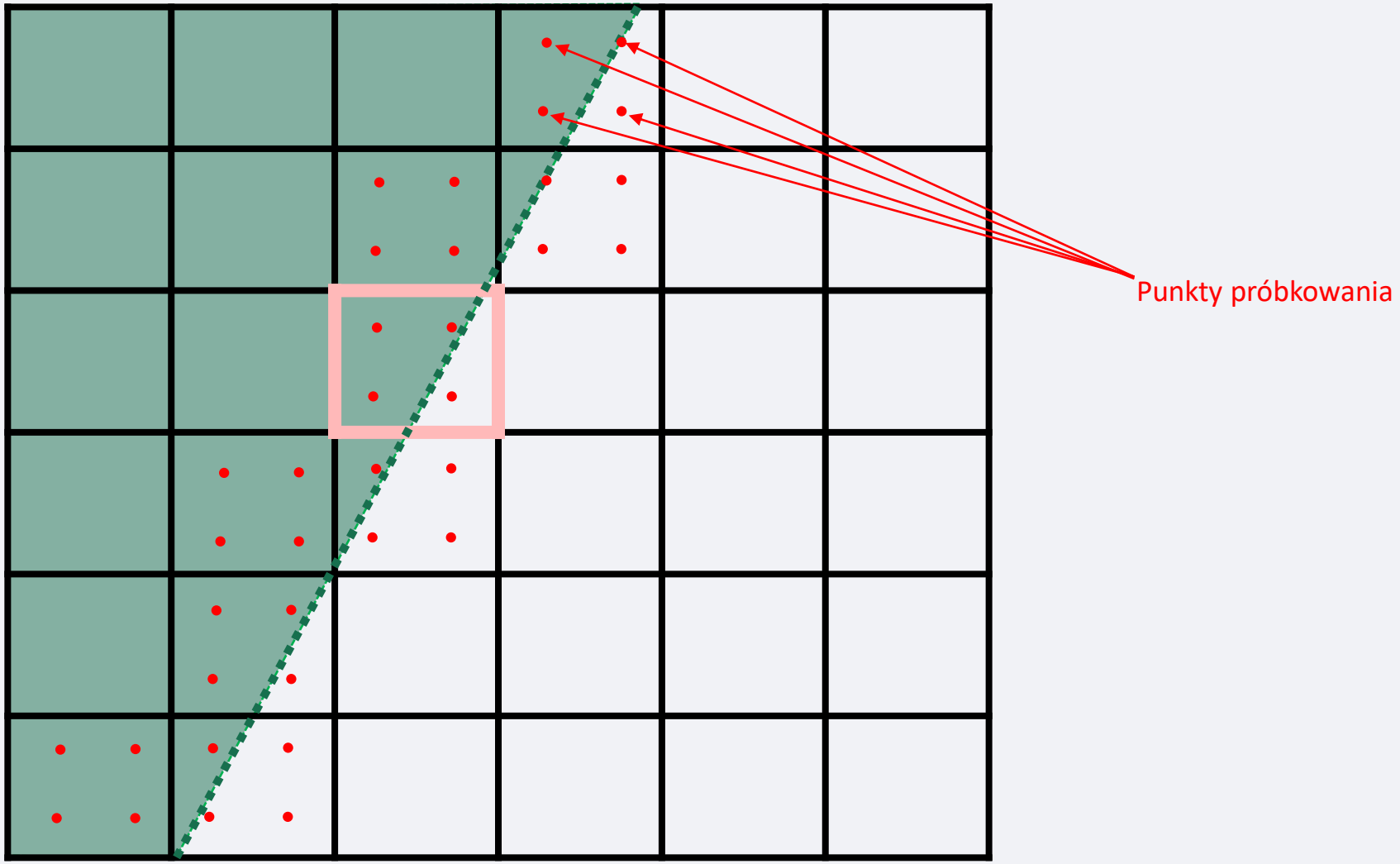
Antyaliasing AA

- *Mutlisamplig / Supersampling* ➔ ograniczenie efektu *aliasingu*
  - *renderowanie* obrazu w rozdzielczości znacznie większej niż wyświetlana (np. 4K ➔ FullHD)
  - próbki kolorów są pobierane w kilku miejscach wewnątrz piksela (a nie tylko jak zwykle w środku)
  - obliczana jest średnia wartość koloru
  - zmniejszanie do żądanego rozmiaru i wykorzystywanie dodatkowych pikseli do obliczeń
- Rezultat
  - ✓ obraz ze zmniejszoną częstotliwością próbkowania (ang. ***downsampled***)
  - ✓ płynniejsze przejścia od jednej do drugiej linii pikseli wzdłuż krawędzi obiektów

Multi-sampling AA



Multi-sampling

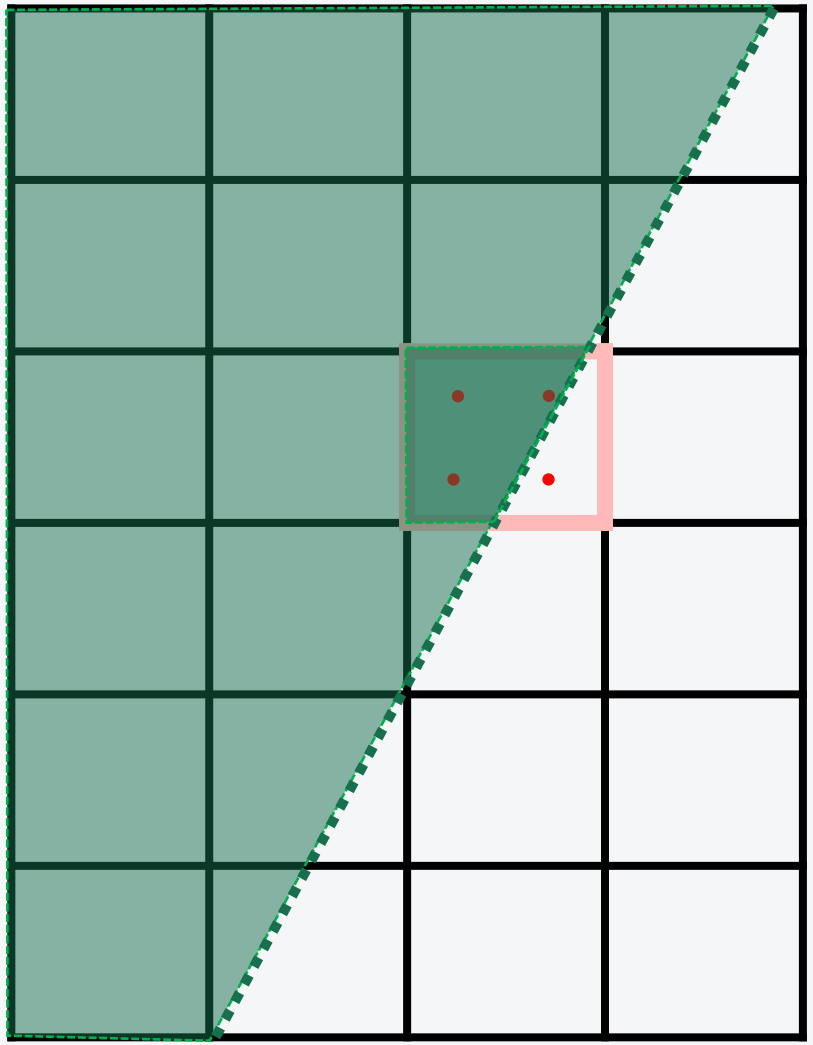


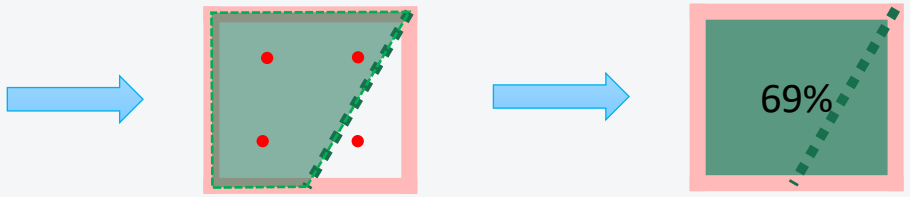
Multi-sampling AA

Multi-sampling AA

Multi-sampling AA

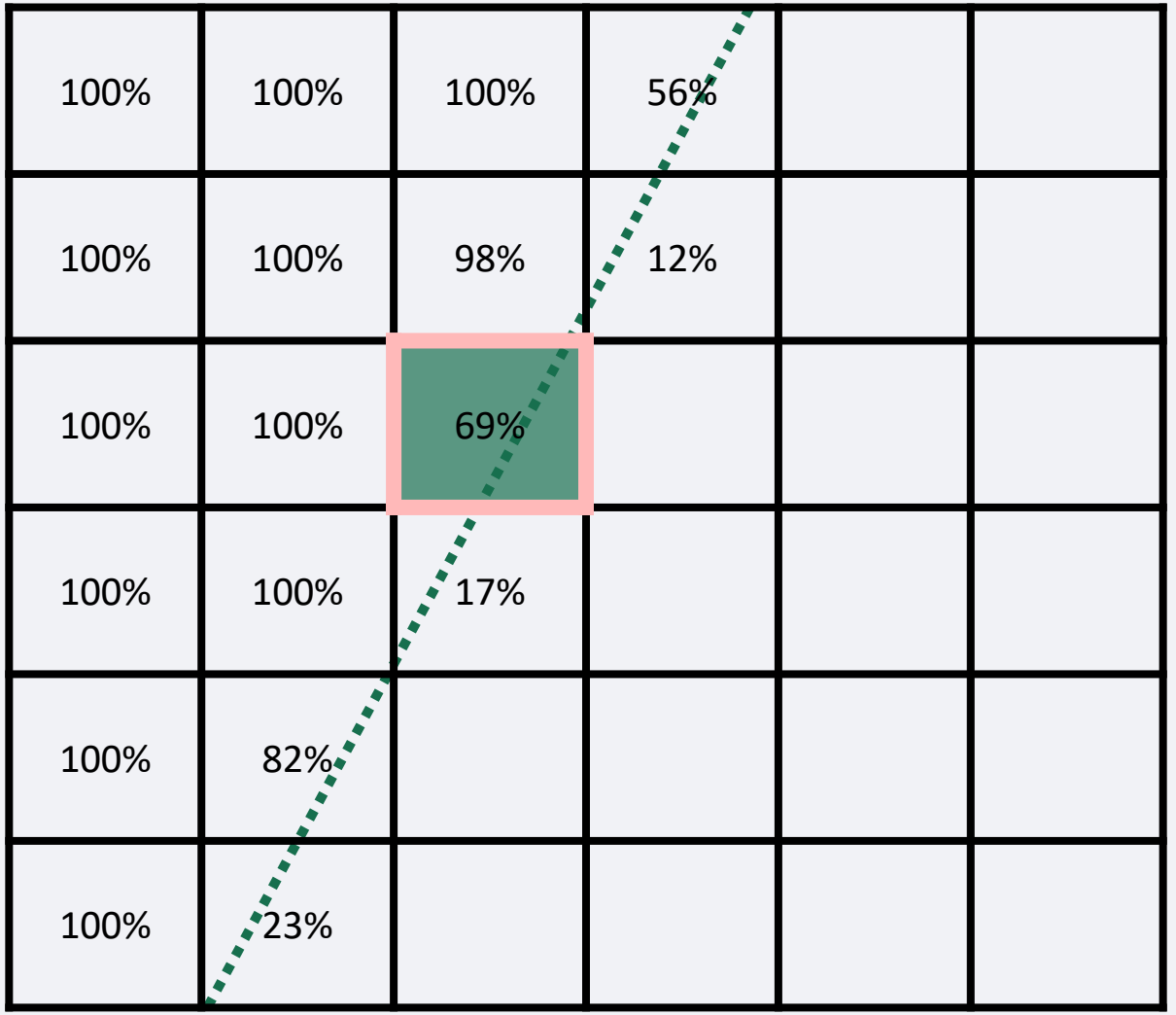
Multi-sampling





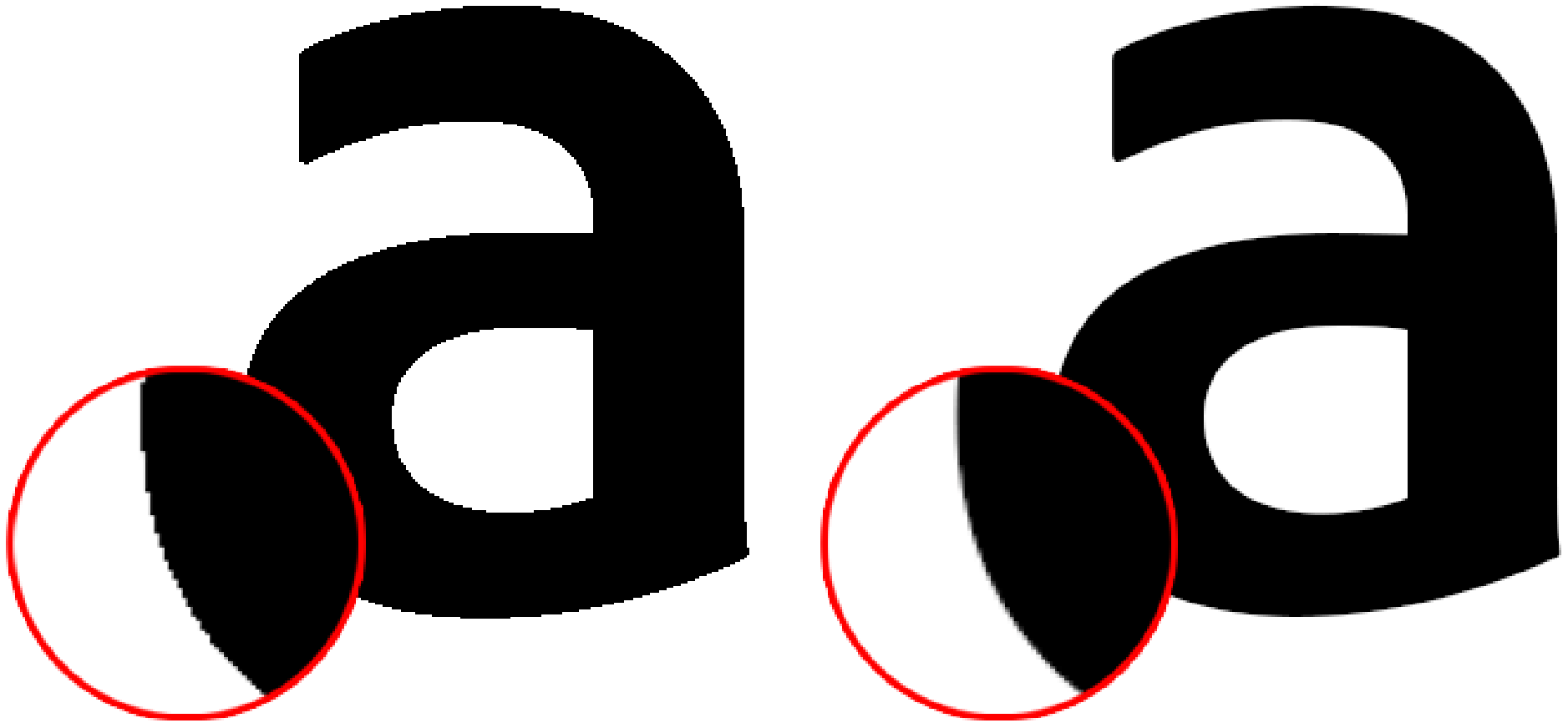
$$px_{kolor} = \frac{kolor\ próbki_1 + kolor\ próbki_2 + \cdots + kolor\ próbki_n}{ilość\ próbek}$$

Multi-sampling AA



Multi-sampling AA

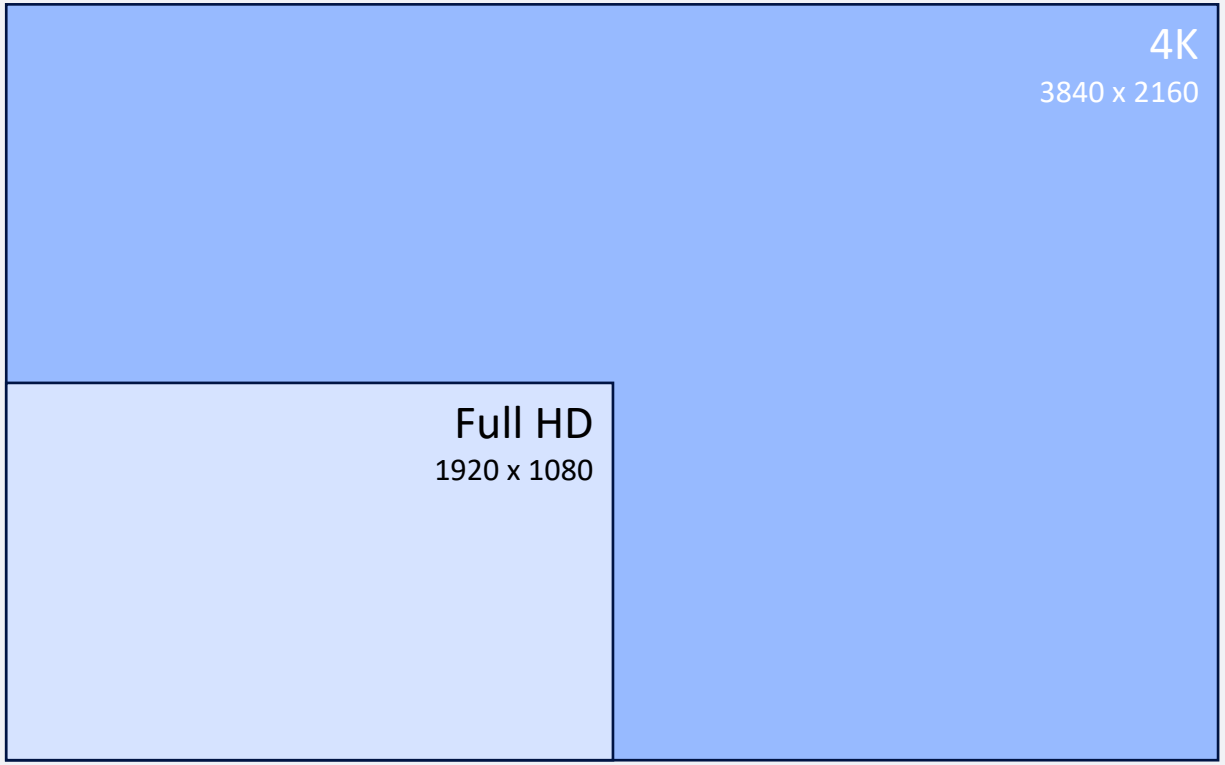
Efekt działania AA

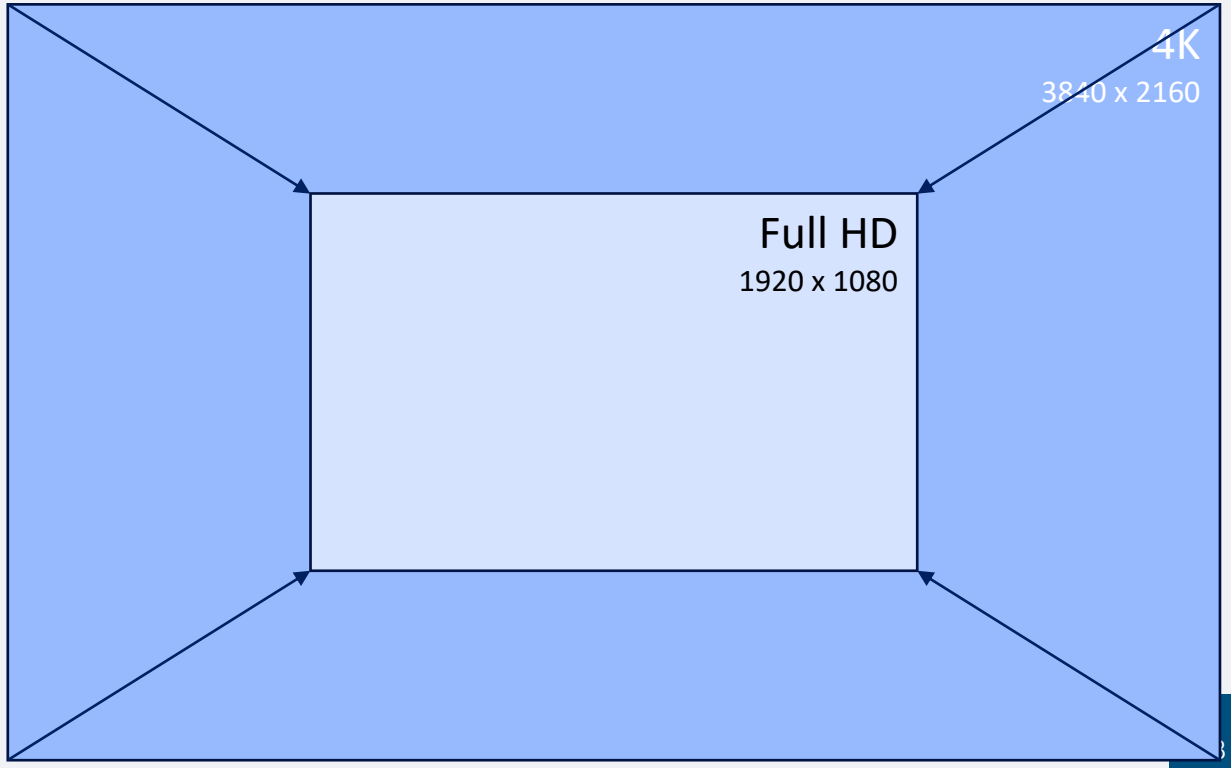


Antialiasing

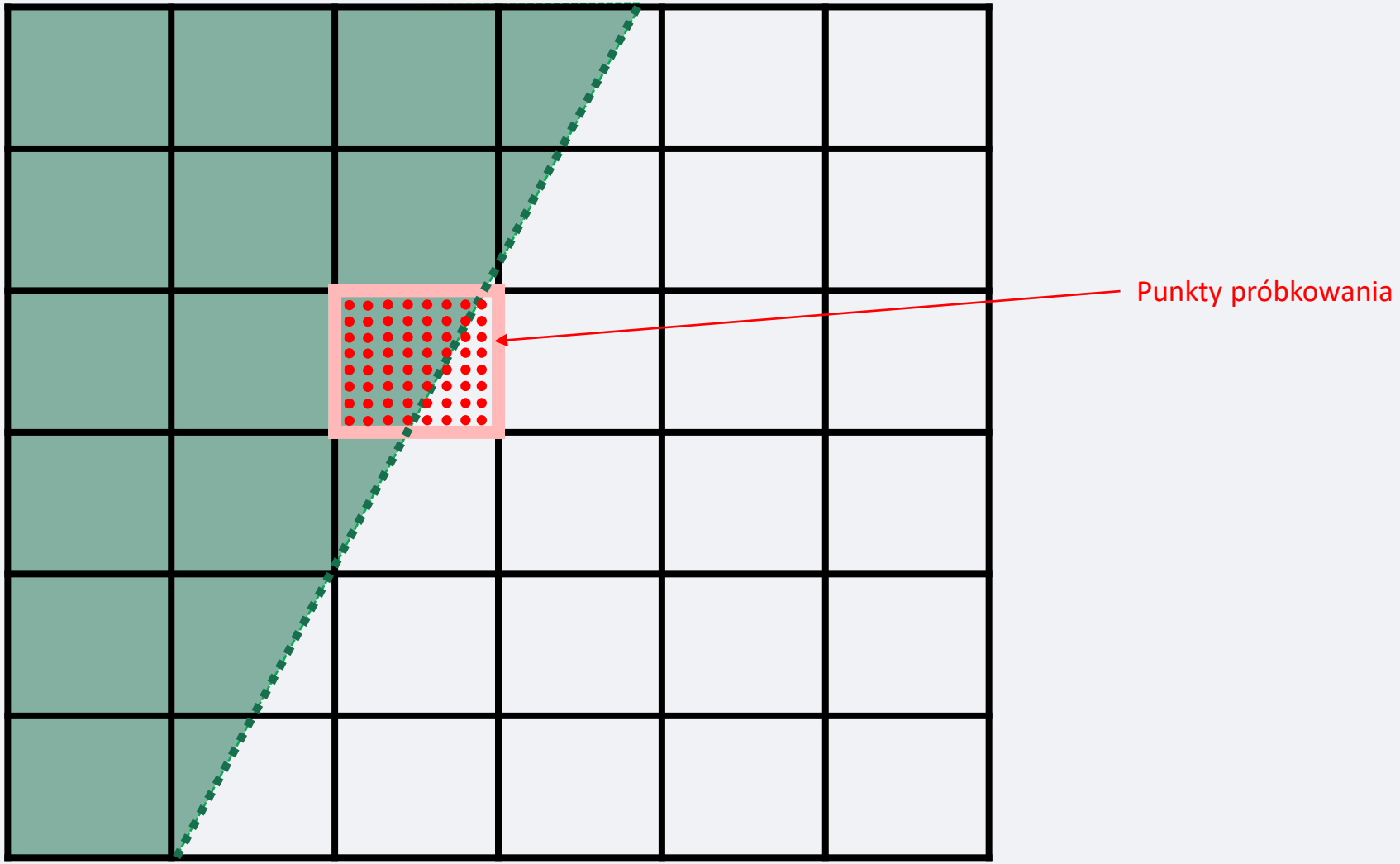
Supersample Anti-Aliasing (SSAA) aka FSAA

- każda klatka jest *renderowana* z rozdzielczością (2-4x)* większą od rozdzielczości monitora, a następnie zmniejszana
- w obrębie każdego piksela próbkowane są liczne miejsca

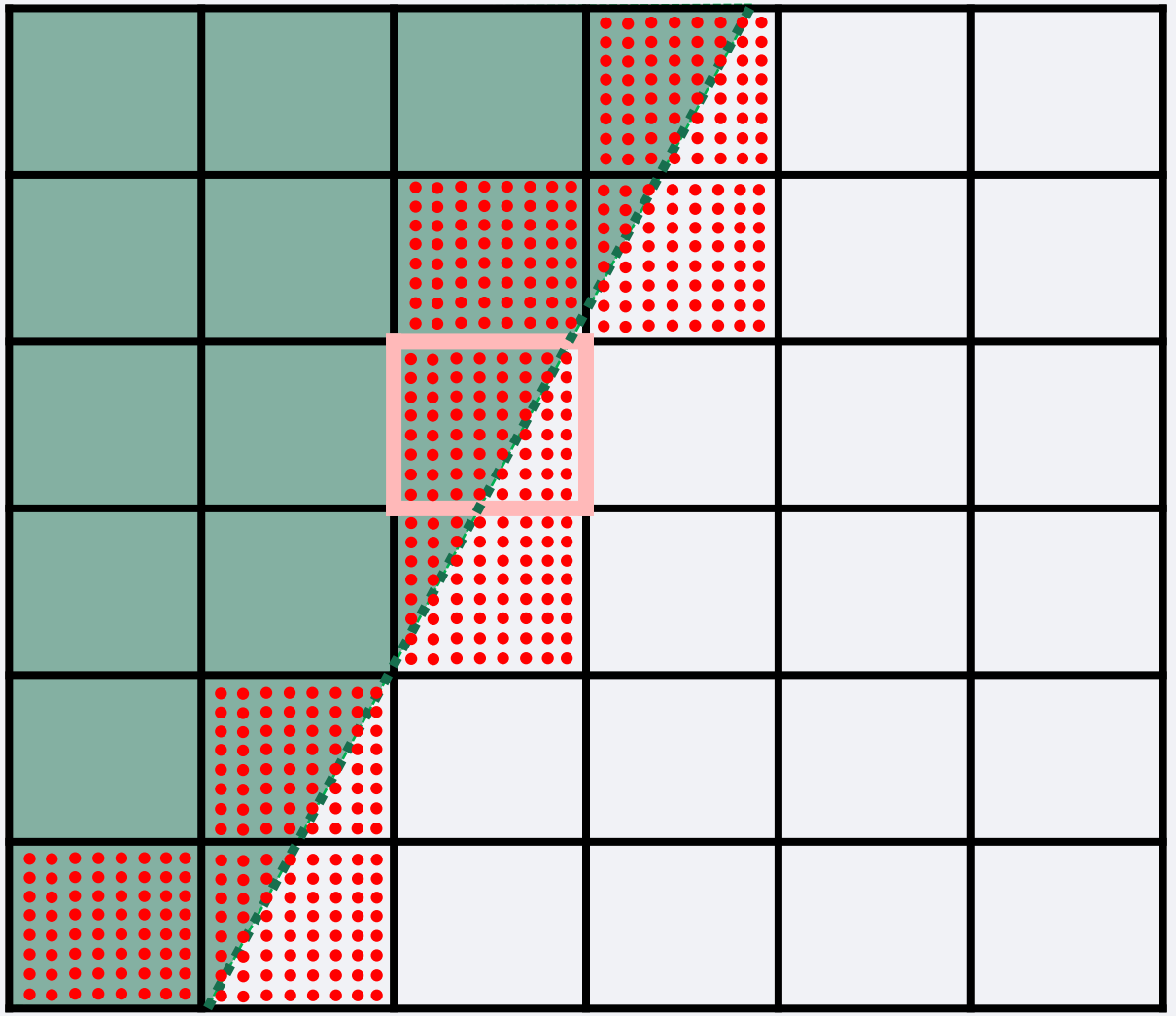




Antialiasing (SSAA)

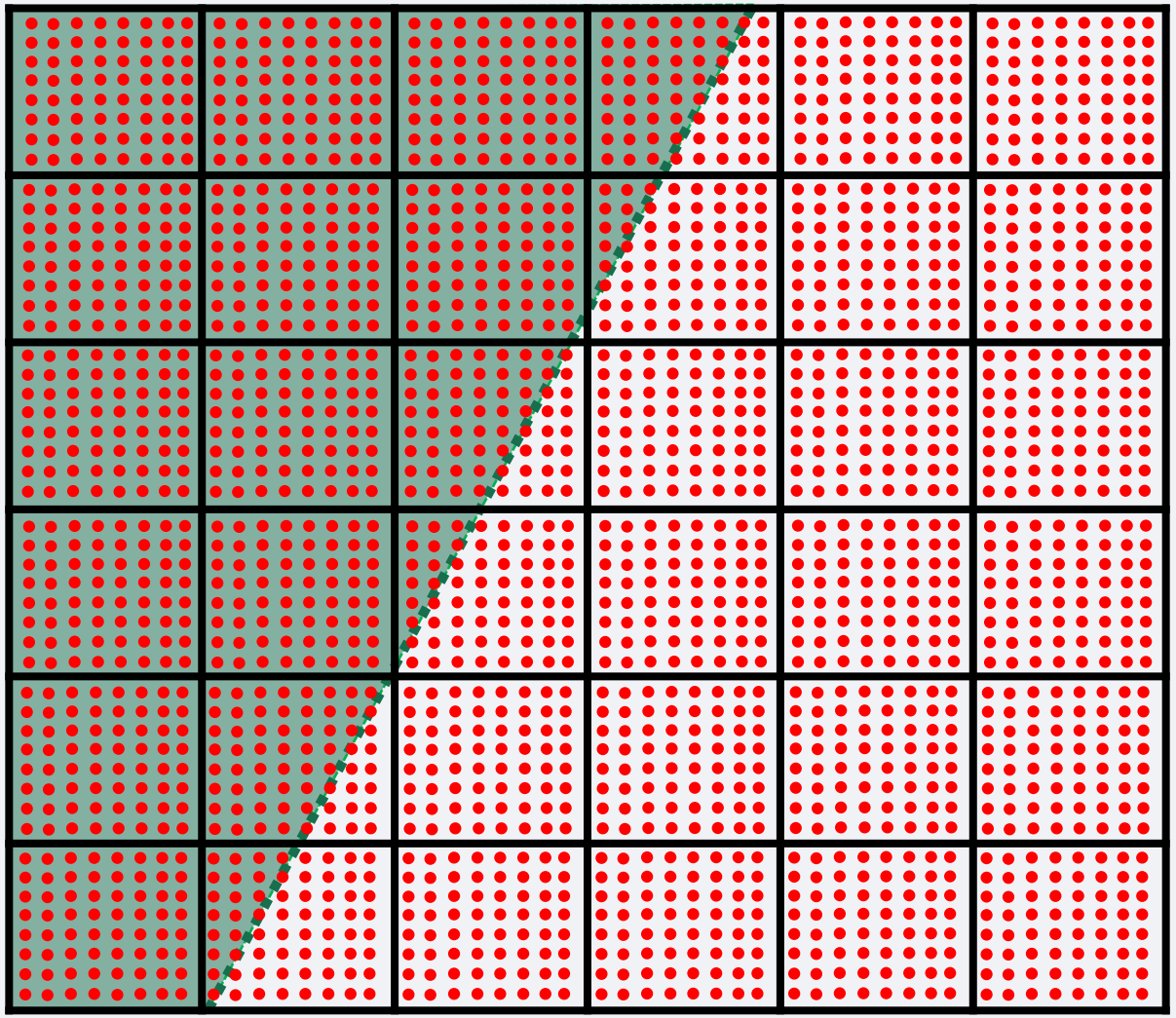


Super-sampling (SSAA)



Punkty próbkowania

Super-sampling (SSAA)



Punkty próbkowania

Antialiasing

Supersample Anti-Aliasing (SSAA)

- każda z tych próbek jest w pełni *renderowana* i łączona z innymi w celu uzyskania piksela
- wykonywany dla każdego piksela ➔ nieefektywne, schodkowanie widoczne w niektórych częściach obrazu, np. na krawędziach

Antialiasing

liczba próbek

- ✓ decyduje o jakości danych wyjściowych
- ✓ większa tym większa alokacja pamięci

kosztowny obliczeniowo różne warianty AA ➔ obniżenie „kosztów”

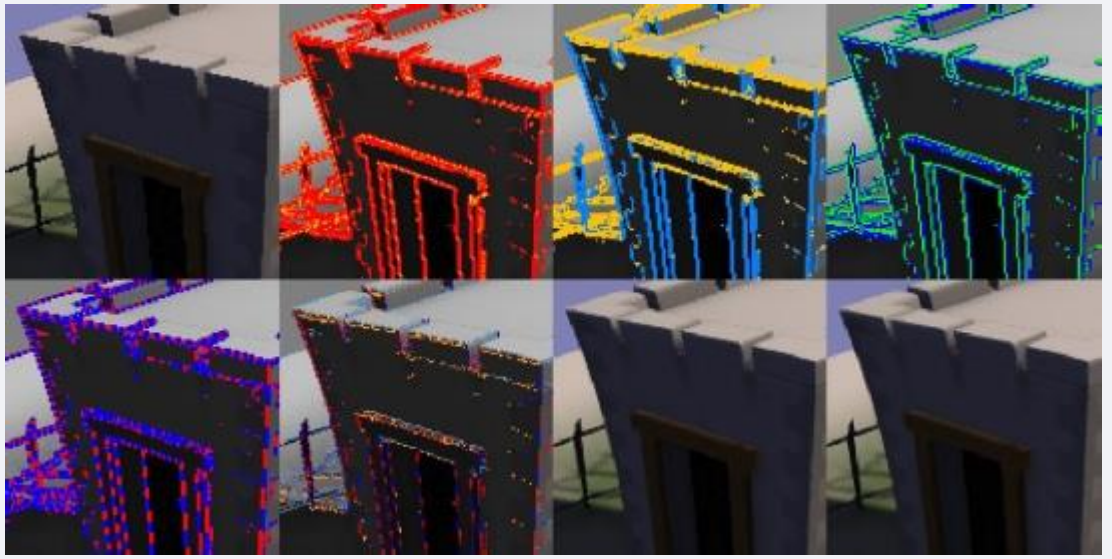
- Adaptive SuperSampling
- Fast Approximate Anti-Aliasing (FXAA)*
- Subpixel Morphological Antialiasing (SMAA)*
- Multisample Frame Anti-Aliasing (MSAA)
- Temporal Anti-Aliasing (TAA)
- *Deep learning super sampling* (*DLSS*)

* Postprocesowy AA

Antialiasing

Fast Approximate Anti-Aliasing (FXAA)

- wygładza krawędzie wszystkich pikseli na ekranie (działa wzdłuż wykrytej krawędzi)
- bazuje na informacji o *luminacji* pikseli
- wykorzystuje filtr górnoprzepustowy ➔ działa na piksele o *wysokim* kontraście
- określa kontrast między sąsiednimi pikselami, heurystycznie znajduje krawędzie i określa czy są poz. / pion.
- obraz bywa zbyt rozmyty
- działa bardzo szybko



https://developer.download.nvidia.com/assets/gamedev/files/sdk/11/FXAA_WhitePaper.pdf

Antialiasing

Morphological antialiasing (MLAA)

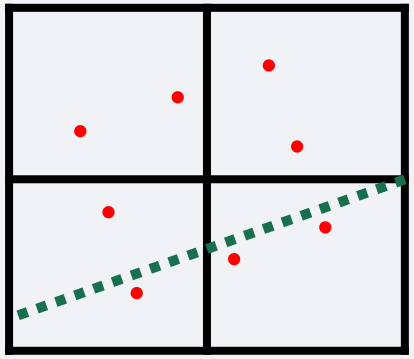
- wykrywa granice na obrazie wynikowym, a następnie znajduje w nich określone wzorce
- mieszanie pikseli na tych granicach, zgodnie z wzorcem, do którego należą, i ich położeniem w obrębie wzorca
- ***Subpixel Morphological AntiAliasing*** (SMAA) wersja *MLAA* opracowana przez Universidad de Zaragoza i firmę Crytek

http://igm.univ-mlv.fr/~biri/mlaa-gpu/TMLAA.pdf

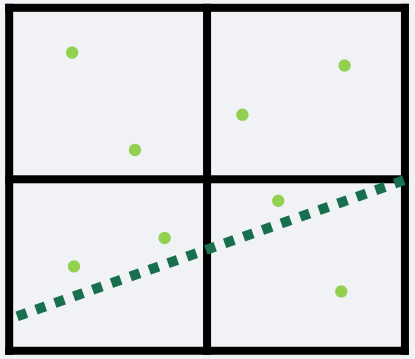
Antialiasing

Multisample Anti-Aliasing (MSAA)

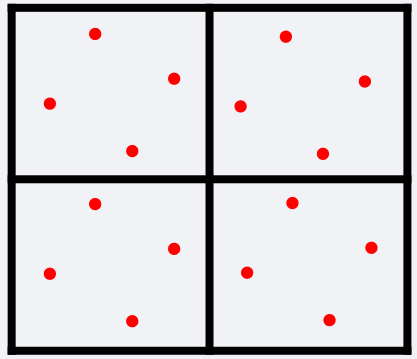
- wykorzystuje algorytmy wykrywania krawędzi do wykrywania aliasingu (na podstawie różnic w kontraście)
- każdy piksel próbkuje się wiele razy (2-8x) w różnych miejscach w kolejnych klatkach (1x) i uśrednia próbki
- piksele próbkowane w poprzednich ramkach są mieszane z pikselami próbkowanymi w bieżącej ramce



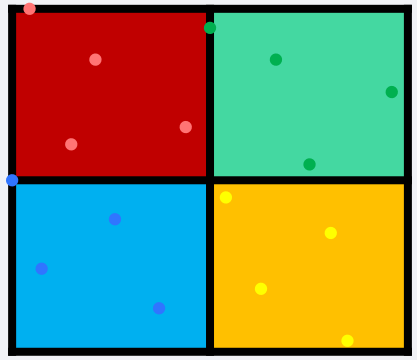
Ramka n-1



Ramka n



Stały wzór 4x



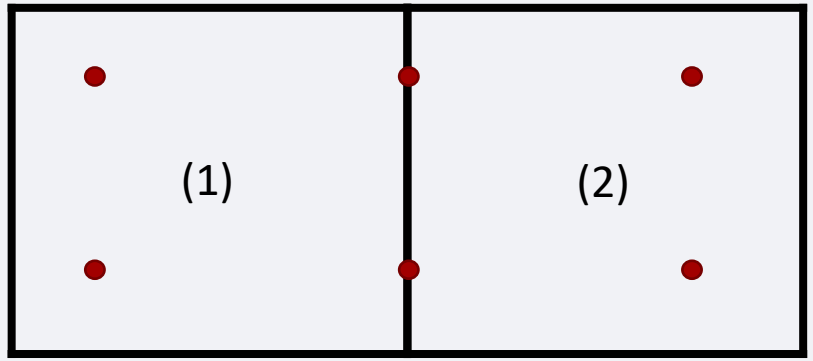
Zmienny wzór 4x

Antialiasing

Multisample Anti-Aliasing (MSAA)

- niektóre próbki mogą być rozdzielone między sąsiadujące piksele (zmniejszenie zapotrzebowania na moc obliczeniową)
- źle radzi sobie z powierzchniami przezroczystymi ➔w podwyższonej rozdzielczości odbywa się jedynie rasteryzacja siatek, a cieniowanie i wypełnianie teksturami przeprowadzane są już w rozdzielczości natywnej

Ramka n



Stały wzór 4x

Zmienny wzór 4x

Antialiasing

Temporal Anti-Aliasing (TAA)

- łączy informacje z bieżącej i z poprzednich klatek
- każdy piksel jest próbkowany 1 / klatkę, ale próbka w innym miejscu w pikselu
- piksele próbkowane w poprzednich ramkach są mieszane z pikselami próbkowanymi w bieżącej ramce
- używa *wektorów ruchu* z silnika gry do wykonania kompensacji ruchu (wygładzanie krawędzi w ruchu)

Antialiasing

DLSS

Cel ➔ uzyskanie wyższej jakości obrazu bez zbytniego pogorszenia framerate'u

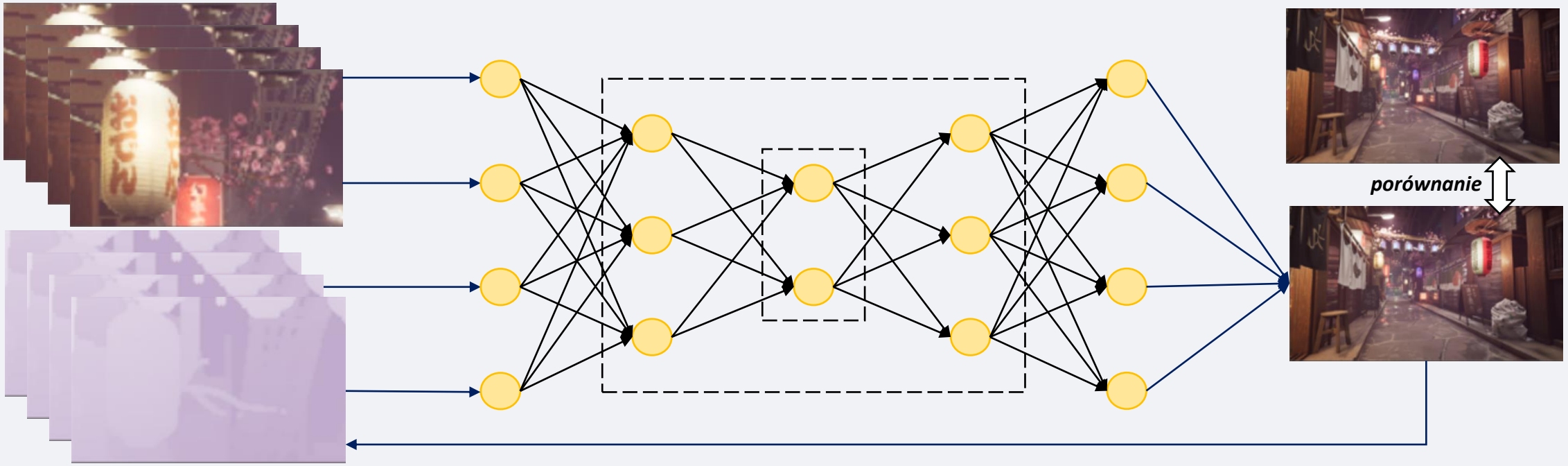
- super próbkowanie z głębokim uczeniem
- zwiększa liczby klatek na sekundę ➔ renderowanie klatek w niższej rozdzielczości niż wyświetlana
- wykorzystanie głębokiego uczenia ➔ przeskalowania klatek tak, aby wyglądały tak samo ostro, jak w natywnej (niższej) rozdzielczości
- klatki renderowane w rozdzielczości 1080p, co pozwala na osiągnięcie wyższej liczby klatek na sekundę, a następnie skalowane i wyświetlane w rozdzielczości 4K, co daje „ostrzejszy” obraz niż w 1080p

DLSS

Autoencoder Variational
(Autoenkoder wariacyjny)

Wejście: *obrazy w „niskiej" rozdzielczości Full HD (aliasing) wyrenderowane przez silnik gry"*

Obraz referencyjny „ultra wysokiej" rozdzielczości 16K (15360 x 8640) wyrenderowany offline



Wejście: Motion Vectors
(niskiej rozdzielczości wektory ruchu z tych samych obrazów - wygenerowane przez silnik gry)

Temporal Feedback
(Czasowe sprzężenie zwrotne)

Wyjście: rozdzielczość 4K eliminacja aliasing'u

DLSS

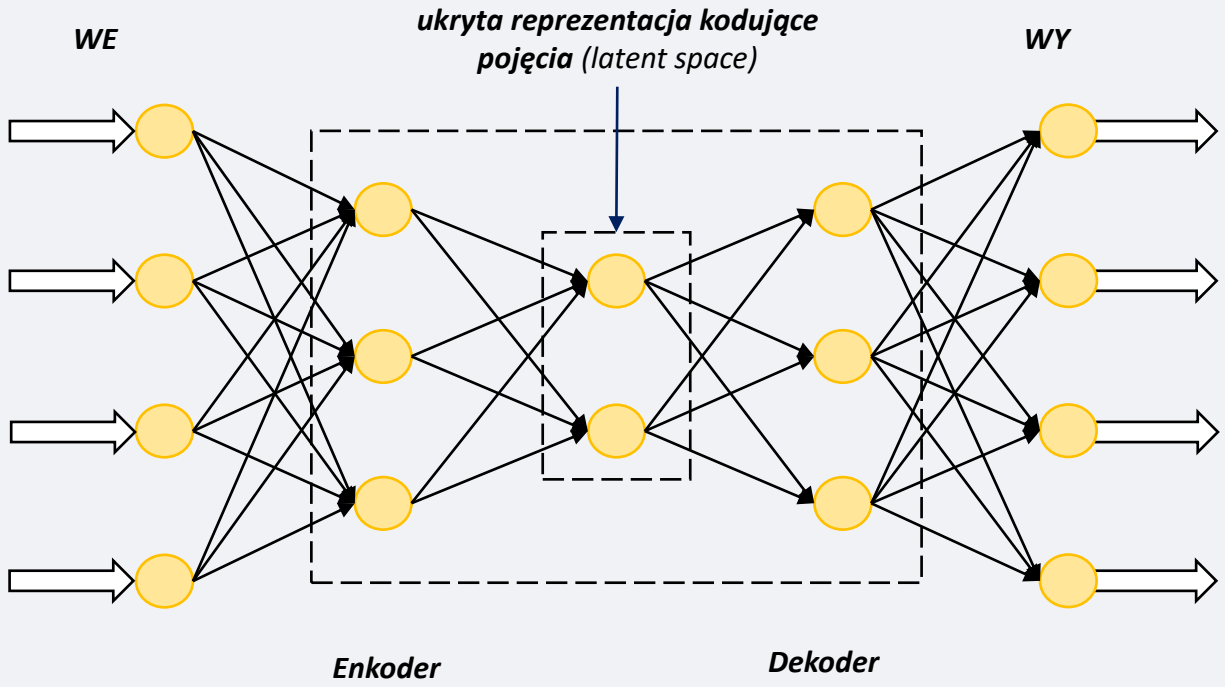
Trening sieci neuronowej DLSS AI zwanej ***variational autoencoder*** na superkomputerze NVIDIA NGX

1. podanie na wejście sieci tysięcy zrzutów ekranu z gry, każdy z 64-krotnym *antyaliasingiem supersamplowanym*
2. podanie obarów bez AA
3. porównanie ujęć ➔ nauka jak „aproksymować jakość" obrazu z antyaliasingiem supersamplowanym 64x, używając klatek źródłowych o niższej jakości

Autoenkoder Wariacyjny (VAE)

Architektura sieci neuronowej wykorzystywana do uczenia bez nadzoru

- sieć jednokierunkowa ➔ celem jest rekonstrukcja sygnału wejściowego
- koder i dekoder są trenowane wspólnie w taki sposób, by dane wyjściowe minimalizowały błąd rekonstrukcji w sensie rozbieżności *Kullbacka-Leiblera*



Dywergencja (odległość) *Kullbacka-Leiblera*

Entropia względna / relatywna

- miara stosowana w statystyce i teorii informacji
- rozbieżność między dwoma rozkładami prawdopodobieństwa $\boldsymbol{p}$ i $\boldsymbol{q}$
- przyjmuje zawsze wartości nieujemne, 0 ⇔ gdy porównywane rozkłady są identyczne

dla rozkładów dyskretnych:

$$\mathbf{d_{KL}(p,q) = \sum_{i} p(i) \log_2 \frac{p(i)}{q(i)}}$$

$\boldsymbol{p}$ - dane rzeczywiste

$\boldsymbol{q}$ - teoretyczny model

Definicje

Epoka (ang. *epoch*)

- ✓ uaktualnienie wag podczas uczenia po wyliczeniu przeciętniej wartości gradientu na podstawie określonej liczby przypadków

Konwolucja (ang. *convolution*)

- ✓ połączenie dwu funkcji dzięki wyliczeniu obszaru w którym będą się pokrywać, kiedy jedna funkcja zostanie nałożona na drugą

Gradient

- ✓ pochodna funkcji, która ma więcej niż jedną zmienną wejściową. matematyki jako nachylenie funkcji
- ✓ mierzy zmianę wszystkich wag w odniesieniu do zmiany błędu

Metoda gradientu prostego (ang. *gradient descent*)

- ✓ metoda optymalizacji, w której parametry są zmieniane w każdej epoce w celu zredukowania funkcji strat, która jest miarą tego, jak dobre wyniki uzyskuje model sieci

Definicje

Funkcja kosztu (ang. *cost function*)

- ✓ funkcja, która precyzuje cel sieci i określa liczbowo wyniki jej uczenia. Celem uczenia sieci jest zminimalizowanie funkcji strat
- ✓ dotyczy pojedynczego przykładu szkoleniowego/wejścia

Funkcja kosztu (ang. *cost function*) **<>** ***Funkcja straty*** (ang. loss function)

Funkcja straty (ang. loss function)

- ✓ określa różnicę między oczekiwanym wynikiem a wynikiem uzyskanym przez model uczenia maszynowego
- ✓ średnia strata dla całego zbioru danych szkoleniowych
- ✓ z niej można wyliczyć *gradient*, wykorzystywany do aktualizacji wag ➔ koszt stanowi średnia ze wszystkich strat

Definicje

Normalizacja (ang. normalization)

- ✓ utrzymanie amplitudy sygnału w określonym zakresie
- ✓ jedną z metod normalizacji zmiennego w czasie dodatniego sygnału jest podzielenie go przez jego wartość maksymalna, dla której granicą jest później 1

Ograniczenia (ang. *constraints*)

- ✓ warunki które musi spełnić rozwiązanie problemu optymalizacji żeby uzyskać pozytywny wynik

Optymalizacja (ang. optimalization)

- ✓ proces wyznaczania wartości maksymalnej / minimalnej funkcji za pomocą systematycznego wyszukiwania wartości wejściowych pochodzących z ich dozwolonego zbioru w celu znalezienie optymalnej wartości funkcji

Definicje

Przeuczenie (ang. overfitting)

- ✓ stan osiągnięty przez algorytm uczenia, w którym liczba regulowanych parametrów w modelu sieci jest znacznie większa niż liczba danych uczących i algorytm zaczyna korzystać z dodatkowej pojemności w celu zapamiętania przypadków. Przeuczenie znacznie ogranicza zdolność do generalizowania zdobytej wiedzy na nowe przypadki. Może ono zostać zredukowane za pomocą *regularyzacji*.

Regularyzacja (ang. *regularization*)

- ✓ metoda pozwalająca na uniknięcie przeuczenia modelu sieci neuronowej z wieloma parametrami, kiedy dane uczące się ograniczone, np. następuje *zanikanie wag* (ang. *weight decay*)
- ✓ polega ona na obniżaniu wag w sieci w każdej kolejnego epoce uczenia, w wyniku czego zostają zachowane tylko wagi i dużych dodatnich gradientach

Definicje

Rozkład prawdopodobieństwa (ang. *probability distribution*)

- ✓ funkcja, która określa prawdopodobieństwo wystąpienie wszystkich możliwych stanów systemu lub wyników eksperymentu

Wsteczna propagacja błędu (ang. *backprop, backpropagation of errors*)

- ✓ algorytm uczący, który optymalizuje sieć neuronowa metodą gradientu prostego w celu zminimalizowania funkcji strat i zwiększenie skuteczności uczenia

Definicje

Zbiory trenująca i testujące (ang. *training and test sets*)

- ✓ wyniki osiągane na zbiorze trenującym nie pozwalają właściwie oszacować, jak sieć poradzi sobie z nowymi przypadkami
- ✓ do oceny, jak dobrze sieć generalizuje zdobytą wiedzę, korzysta się ze zbioru testującego nieużywanego podczas uczenia
- ✓ kiedy zbiory danych są małe, używa się pojedynczej próbki wyodrębnionej ze zbioru testującego., dzięki czemu można sprawdzić wyniki uczenia przeprowadzonego na pozostałych przypadkach z tego zbioru.
- ✓ proces powtarza się dla każdej próbki przy czym rezultaty są uśredniane w celu uzyskania jednego wyniku, Jest to szczególny przypadek walidacji krzyżowej (ang. *cross validation*), w którym liczba elementów podzbioru testowego n wynosi 1

Generatywne sieci współzawodniczące

- Ian Goodfellow (2014)
- dwie sieci neuronowe rywalizują ze sobą w grze (o sumie zerowej ➔ zysk jednego agenta jest stratą drugiego)
- GAN (Generative Adversarial Networks)
  - ✓ Generatywne sieci współzawodniczące
  - ✓ Generatywne sieci przeciwstawne
  - ✓ Generatywne sieci adwersaryjne

https://this-person-does-not-exist.com

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., 2014. Generative adversarial nets. *Advances in neural information processing systems*, *27*.

Generatywne sieci współzawodniczące

- Generatywne sieci współzawodniczące (ang. ***Generative Adversarial Networks***), to podejście do modelowania generatywnego z wykorzystaniem metod głębokiego uczenia, takich jak neuronowe sieci konwolucyjne
- GAN-y to sposób trenowania modelu generatywnego, polegający na przedstawieniu problemu jako problemu uczenia nadzorowanego z dwoma modelami:
  - modelem **generatywnym**, który trenujemy do generowania nowych przykładów
  - modelem **dyskryminacyjnym**, klasyfikuje przykłady jako prawdziwe (z dziedziny) lub fałszywe (wygenerowane)

Generatywne sieci współzawodniczące

- pierwotnie zaproponowane jako model generatywny do
  - uczenia bez nadzoru (ang. *unsupervised learning*)
- przydatne do
  - uczenia częściowo nadzorowanego (ang. *semi-supervised learning*)
  - w pełni nadzorowanego (ang. *fully supervised learning*)
  - uczenia przez wzmocnienie (ang. *reinforcement learning*)

Generatywne sieci współzawodniczące

model generatywny

- ✓ po przejściu przez proces uczenia jest w stanie generować nowe próbki wejściowe
- ✓ opisuje sposób generowania zestawu danych z punktu widzenia modelu probabilistycznego
- ✓ poprzez próbkowanie z tego modelu jesteśmy w stanie wygenerować nowe dane
- ✓ używany, gdy mamy pojęcie o podstawowym rozkładzie danych i chcemy znaleźć ukryte parametry rozkładu

zestaw danych

- ✓ wiele próbek podmiotu, który staramy się wygenerować
- ✓ zestaw próbek ➔ dane szkoleniowe

Generatywne sieci współzawodniczące

obserwacja

- ✓ pojedynczy egzemplarz zbioru danych
- ✓ składa się z wielu cech (ang. *features*)
- ✓ w przypadku obrazów cechami są zwykle poszczególne wartości pikseli

przestrzeń próbek

- ✓ zestaw wszystkich wartości, które może przyjmować *obserwacja* $\boldsymbol{x}$

funkcja gęstości prawdopodobieństwa

- ✓ funkcja $\boldsymbol{p(x)}$ odwzorowuje *obserwację* $\boldsymbol{x}$ należącą do przestrzeni próbek na liczbę z przedziału od $[\mathbf{0}; \mathbf{1}]$
- ✓ suma funkcji gęstości dla wszystkich *obserwacji* w przestrzeni próbek musi być równa 1
- ✓ ściśle zdefiniowany rozkład prawdopodobieństwa

Generatywne sieci współzawodniczące

przestrzeń ukryta (ang. *latent space*)

- ✓ przestrzeń cech
- ✓ służy do opisania każdej obserwacji w zbiorze treningowym
- ✓ abstrakcyjna wielowymiarowa przestrzeń zawierającej wartości cech, których nie możemy bezpośrednio zinterpretować, która koduje znaczącą wewnętrzną reprezentację obserwacji

funkcja mapowania

- ✓ przekształca punkty w *przestrzeni ukrytej* na punkt we właściwej domenie ➔ każdy punkt w przestrzeni ukrytej jest reprezentacją pewnego wielowymiarowego obrazu

Generatywne sieci współzawodniczące

Generator

- Model, który służy do generowania nowych, prawdopodobnych przykładów z dziedziny, której dotyczy problem

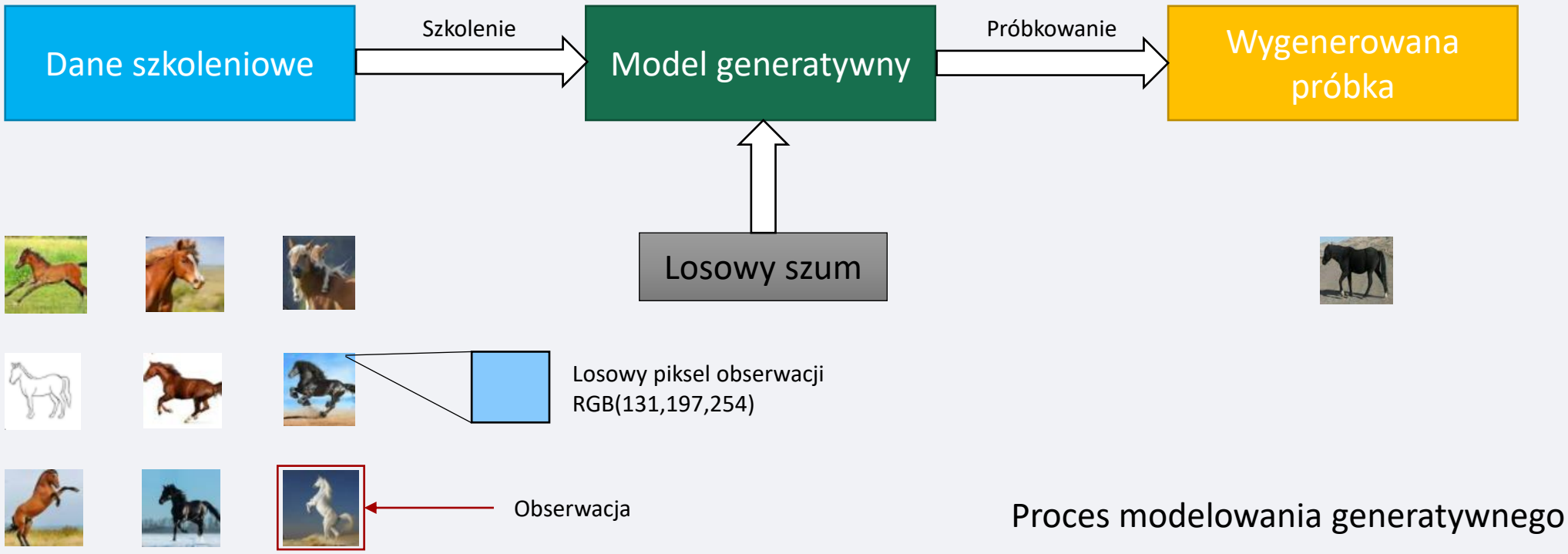
Dyskryminator

- Model służący do klasyfikowania przykładów jako prawdziwych (z domeny) lub fałszywych (wygenerowanych)

Modele są trenowane razem w grze o sumie zerowej

- przeciwstawia się sobie przeciwników $\boldsymbol{D}$ i $\boldsymbol{G}$
- dopóki model $\boldsymbol{D}$ nie zostanie oszukany mniej więcej w połowie przypadków ➔ model $\boldsymbol{G}$ generuje wiarygodne przykłady

Generatywne sieci współzawodniczące



Proces modelowania generatywnego

Foster, David. Generative deep learning: teaching machines to paint, write, compose, and play. O'Reilly Media, 2019.

Generatywne sieci współzawodniczące

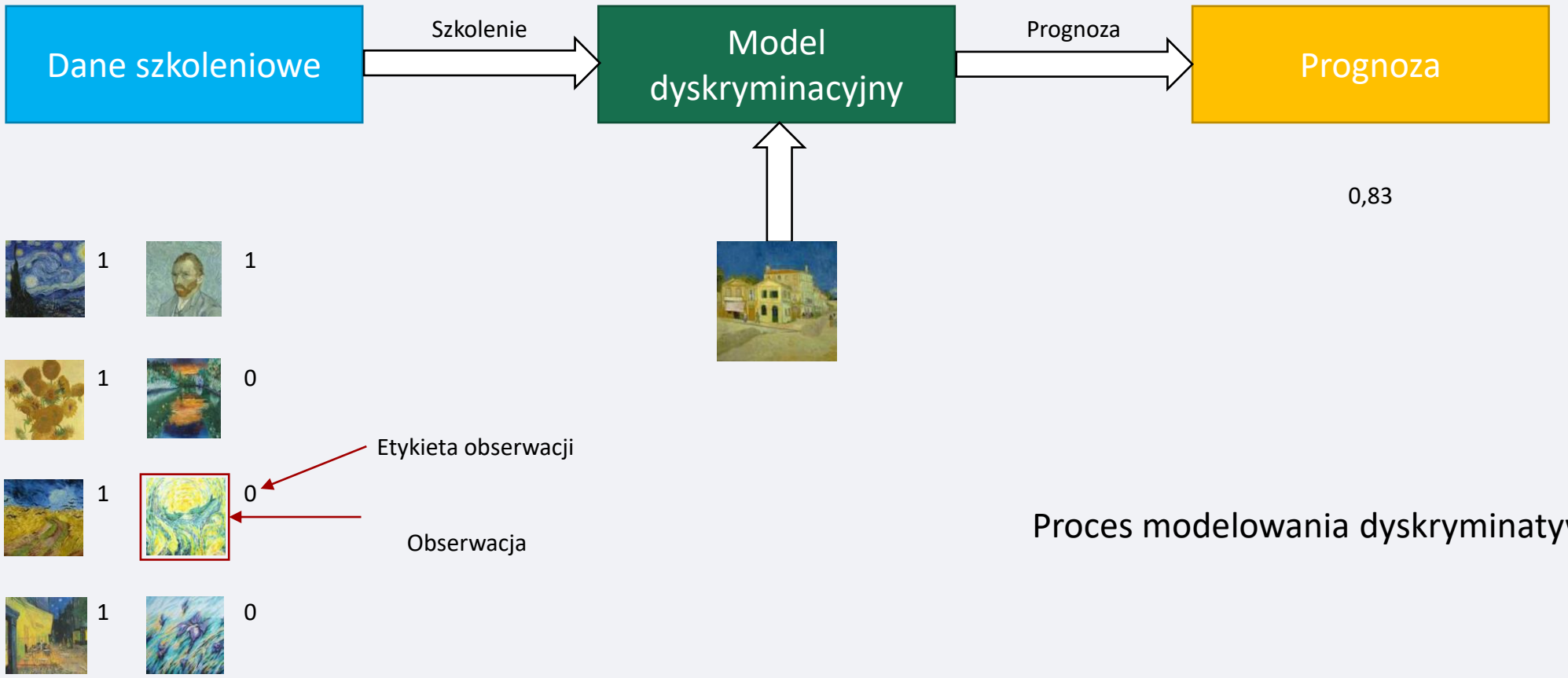
Generator

- przyjmuje jako dane wejściowe wektor losowy o stałej długości i generuje obraz w dziedzinie
- wektor ten jest losowany z rozkładu gaussowskiego (zwanego przestrzenią ukrytą ang. ***latent space***)
- wektor ten jest wykorzystywany do zapoczątkowania procesu generowania
- po treningu model generatora jest przechowywany i wykorzystywany do generowania nowych próbek

Modelowanie generatywne

- szacuje $\boldsymbol{p(x)}$ ➔ prawdopodobieństwo zaobserwowania obserwacji $\boldsymbol{x}$
- jeśli zestaw danych jest oznaczony, możemy również zbudować model generatywny, który oszacowuje rozkład $\boldsymbol{p(x|y)}$

Generatywne sieci współzawodniczące



Proces modelowania dyskryminatywnego

Foster, David. Generative deep learning: teaching machines to paint, write, compose, and play. O'Reilly Media, 2019.

Generatywne sieci współzawodniczące

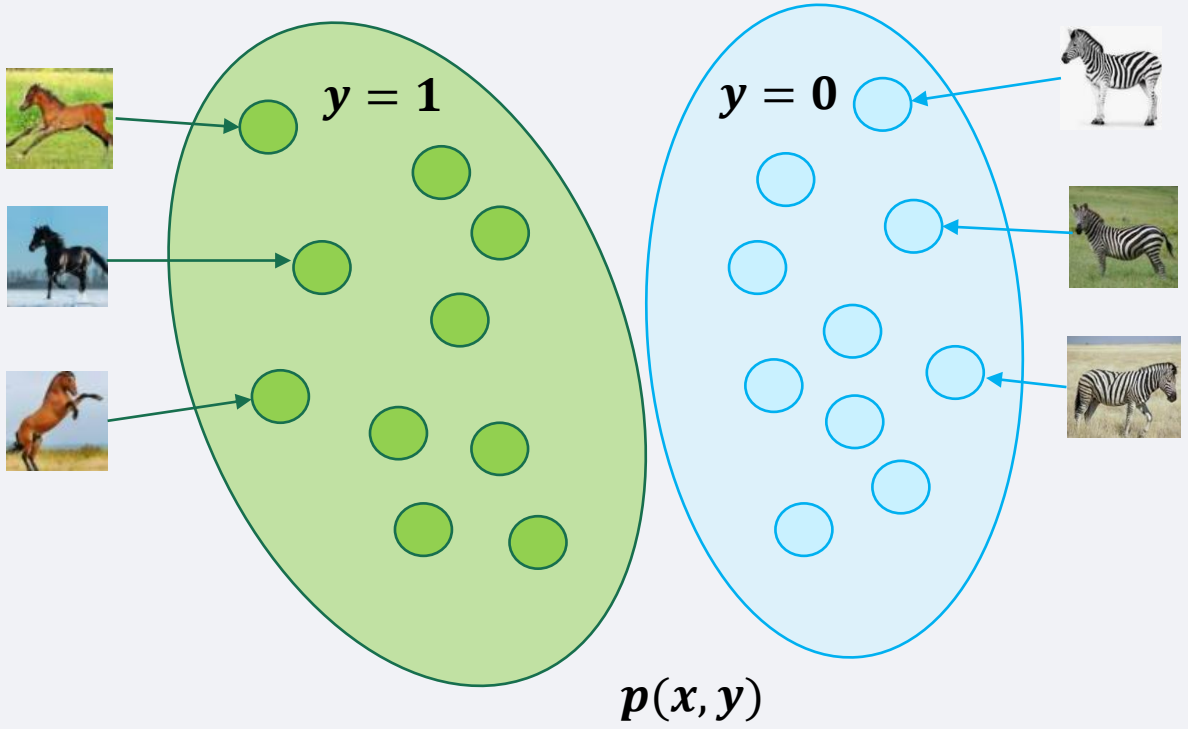
Dyskryminator

- przyjmuje przykład z dziedziny jako dane wejściowe (rzeczywisty lub wygenerowany)
- przewiduje binarną etykietę klasy: prawdziwy lub fałszywy (wygenerowany)
- przykład rzeczywisty pochodzi z zestawu szkoleniowego
- przykłady wygenerowane są wyprowadzane przez model generatora
- jest zwykłym (i dobrze rozumianym) modelem klasyfikacyjnym
- po zakończeniu procesu szkolenia model dyskryminatora jest odrzucany, ponieważ interesuje nas generator

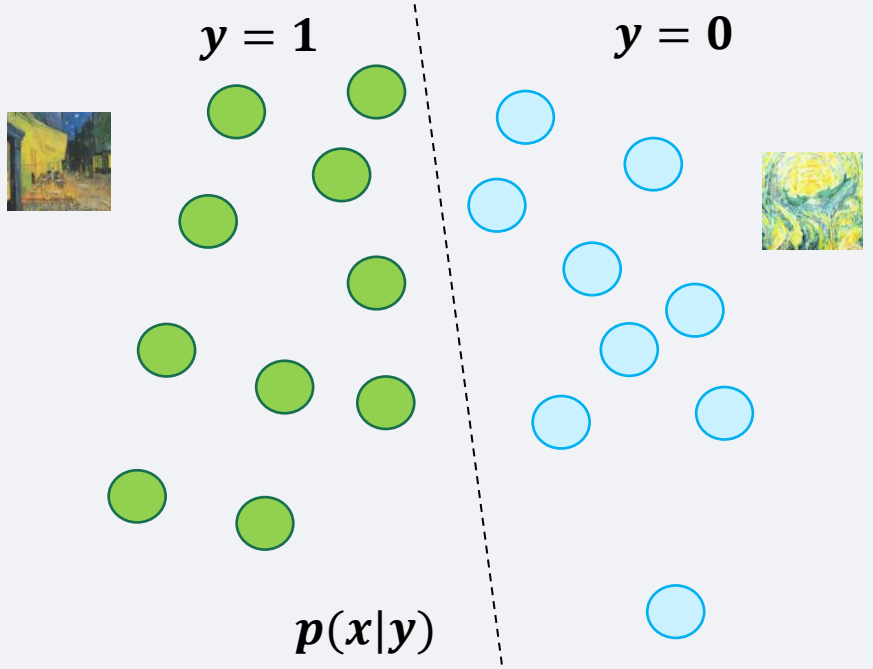
Modelowanie dyskryminatywne

- szacuje wartość $\boldsymbol{p}(\boldsymbol{y}|\boldsymbol{x})$ – prawdopodobieństwo, że danej obserwacji $\boldsymbol{x}$ zostanie przypisana etykieta $\boldsymbol{y}$

Generatywne sieci współzawodniczące



Model generatywny

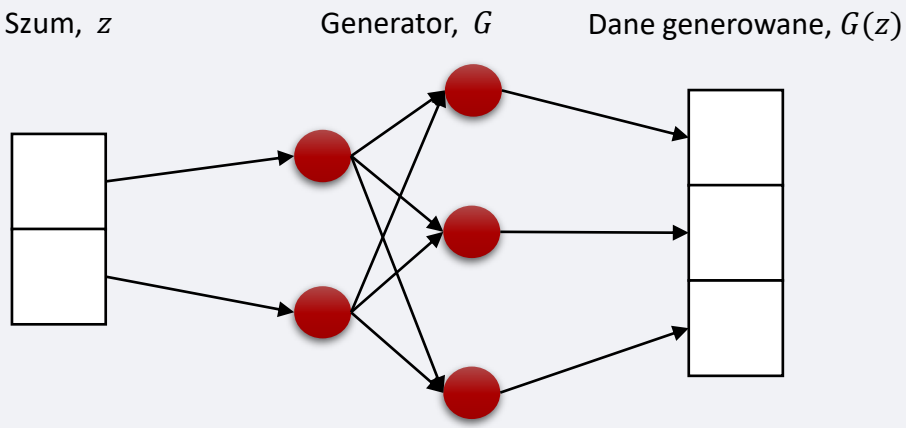


Model dyskryminatywny

Generatywne sieci współzawodniczące

Sieć generująca

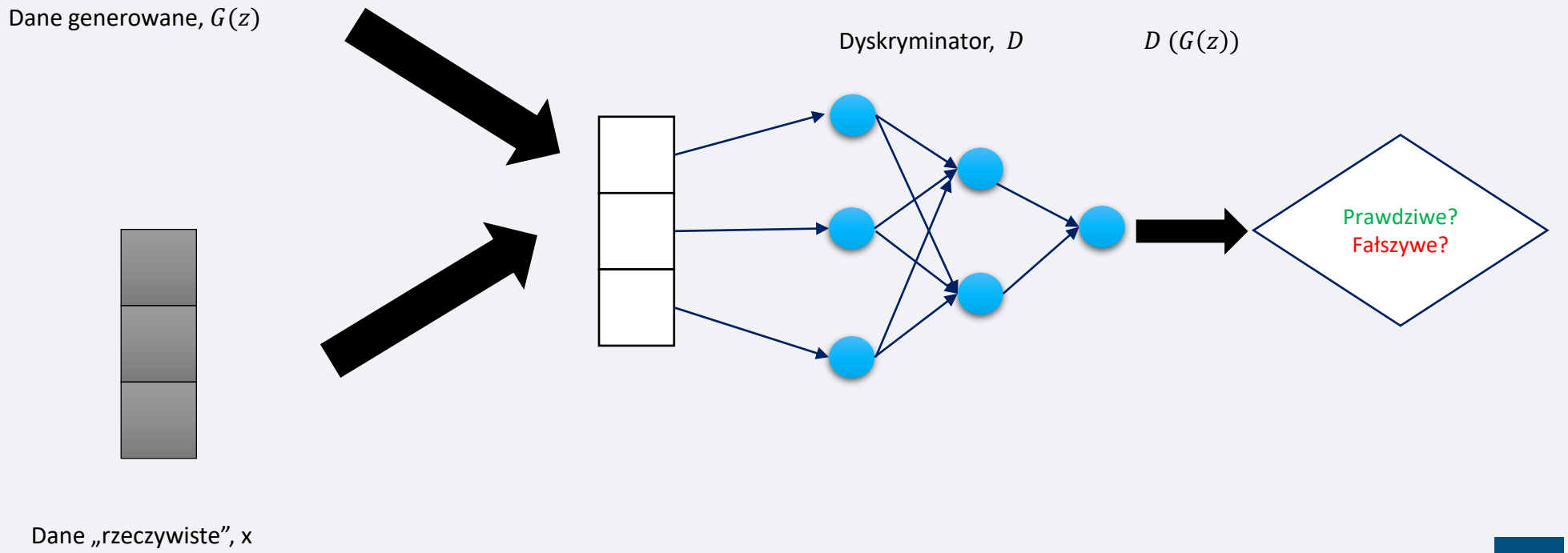
- tworzy sygnał z „pewnego” rozkładu starając się „oszukać” sieć oceniającą
- trening dąży do maksymalizacji błędu dyskryminacji



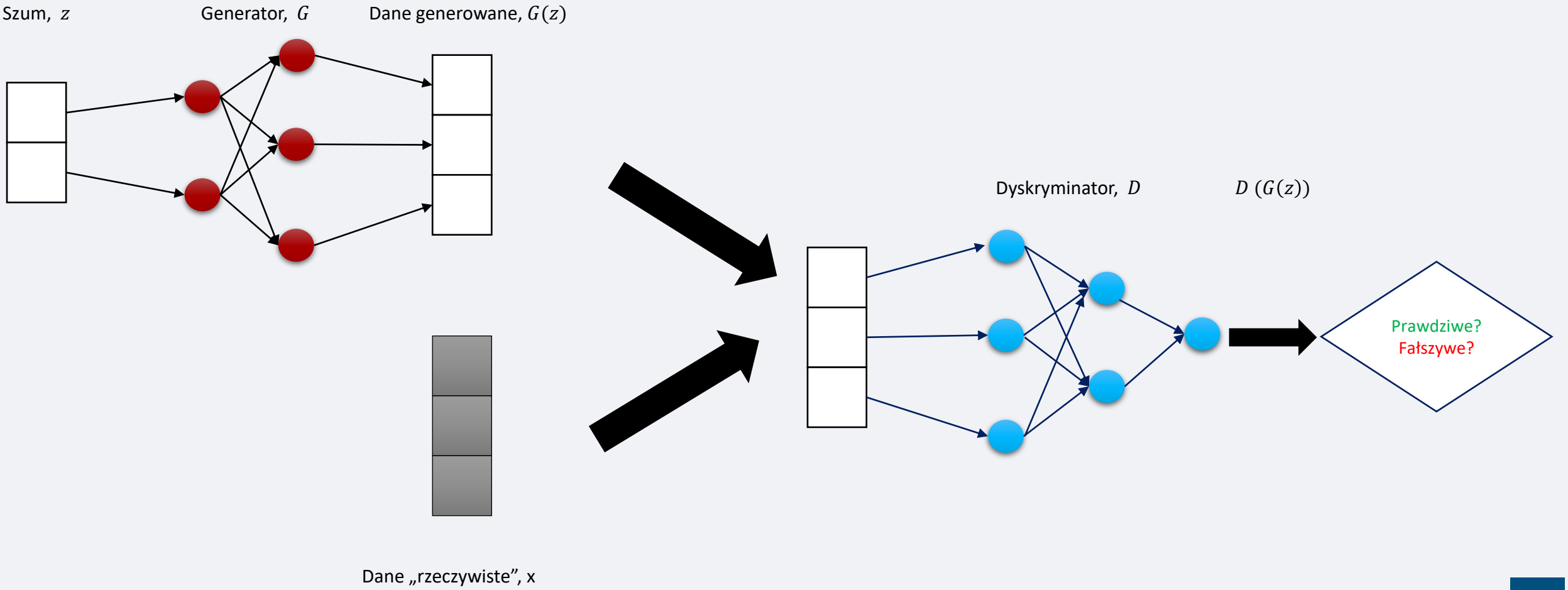
Generatywne sieci współzawodniczące

Sieć oceniająca (dyskryminująca, np. CNN)

- stara się odróżnić prawdziwy sygnał od wygenerowanego przez sieć generującą



Generatywne sieci współzawodniczące



Generatywne sieci współzawodniczące

Funkcja kosztu dyskryminatora

- binarna entropia krzyżowa (spodziewana odpowiedź dyskryminatora dla obrazu prawdziwego = 1)
- trening dąży do maksymalizacji błędu dyskryminacji

$$L_D = -\frac{1}{2}\mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] - \frac{1}{2}\mathbb{E}_{z\sim p_z(z)}\left[\log\left(1 - D\left(G(z)\right)\right)\right]$$

gdzie:

$\boldsymbol{p_{data}(x)}$ rozkład danych prawdziwych

$\boldsymbol{p_z(z)}$ rozkład danych generowanych

Generatywne sieci współzawodniczące

Funkcja kosztu generatora

- maksymalizuje prawdopodobieństwo popełnienia błędu przez dyskryminator
- spodziewana odpowiedź 1 dyskryminatora dla obrazu fałszywego

$$L_G = -\frac{1}{2}\mathbb{E}_{z\sim p_z(z)}\left[\log\left(1 - D\left(G(z)\right)\right)\right]$$

Generatywne sieci współzawodniczące

Trening sieci GAN

- generator $\boldsymbol{G}$ i dyskryminator $\boldsymbol{D}$, są trenowane razem
- pojedynczy cykl treningowy obejmuje najpierw wybór serii obrazów rzeczywistych z dziedziny, której dotyczy problem
- następnie generowana jest partia punktów ukrytych i podawana do modelu $\boldsymbol{G}$ w celu zsyntetyzowania partii obrazów
- $\boldsymbol{D}$ jest następnie aktualizowany przy użyciu partii obrazów rzeczywistych $\boldsymbol{D}$ $(\boldsymbol{x})$ i wygenerowanych $\boldsymbol{G}(\boldsymbol{z})$, minimalizując binarną stratę entropii krzyżowej
- $\boldsymbol{G}$ jest następnie aktualizowany za pośrednictwem modelu dyskryminatora ➔ wygenerowane obrazy są prezentowane w dyskryminatorze tak, jakby były prawdziwe (nie wygenerowane), a błąd jest propagowany z powrotem przez model generatora
- skutkuje to aktualizacją modelu $\boldsymbol{G}$ w kierunku generowania obrazów, co do których istnieje większe prawdopodobieństwo, że oszukają $\boldsymbol{D}$
- proces ten jest następnie powtarzany przez określoną liczbę iteracji treningowych

Generatywne sieci współzawodniczące

Trening sieci GAN (pojedyncza iteracja)

for each training iteration **do**

Sieć oceniająca (dyskryminująca)

for k steps **do**

sample m **noise** samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_g(z)$ and transform with ***Generator***

sample m **real** samples $\{x^{(1)}, \ldots, z^{(m)}\}$ from real data (from data generating distribution) $p_{data}(x)$

update ***Discriminator*** by ***ascending*** the stochastic gradient

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(x^{(i)}\right) + \log\left(1 - D\left(G\left(z^{i}\right)\right)\right) \right]$$

end for

Sieć generująca

sample m **noise** samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_g(z)$ and transform with ***Generator***

update ***Generator*** by ***descending*** the stochastic gradient

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \left[\log\left(1 - D\left(G\left(z^{i}\right)\right)\right) \right]$$

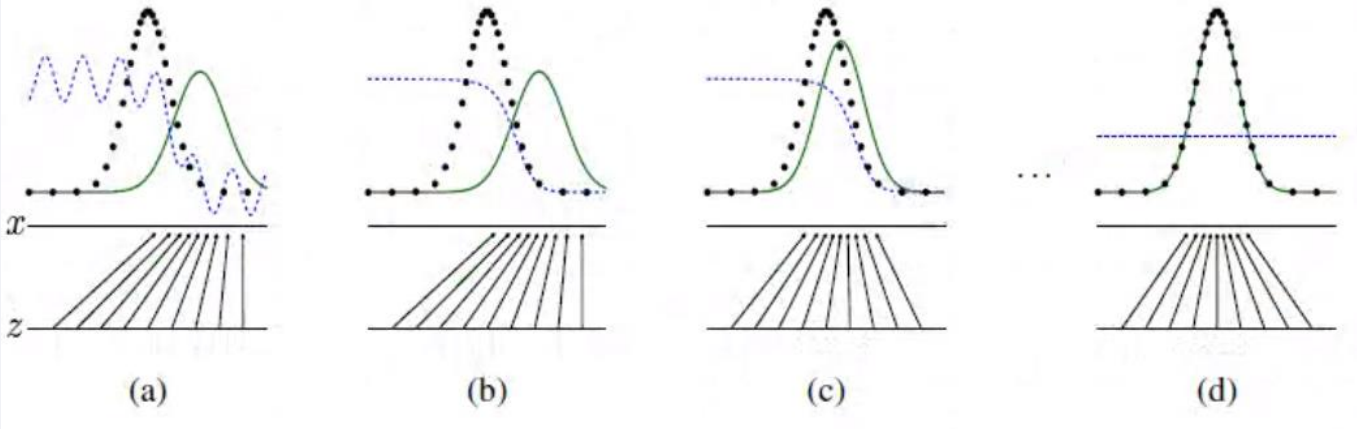
end for

Generatywne sieci współzawodniczące

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}\left[\log\left(1 - D\left(G(z)\right)\right)\right]$$

Wyjście dyskryminatora dla danych rzeczywistych x

Wyjście dyskryminatora dla wygenerowanych fałszywych danych G(z)



Dolna linia - obszar, z którego próbkowane jest równomiernie z

Strzałki pokazują, jak odwzorowanie $\boldsymbol{x} = \boldsymbol{G}(\boldsymbol{z})$ nakłada rozkład niejednorodny p_g na przekształcone próbki

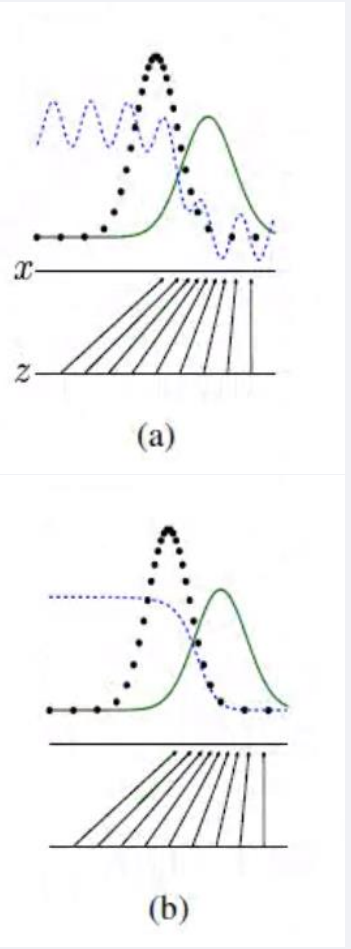
G kurczy się w regionach o dużej gęstości i rozszerza w regionach o małej gęstości p_g

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., 2014. Generative adversarial nets. *Advances in neural information processing systems*, *27*.

Generatywne sieci współzawodniczące

(a) rozważmy parę przeciwników bliską zbieżności: p_g jest podobne do p_{data}, a D jest częściowo dokładnym klasyfikatorem

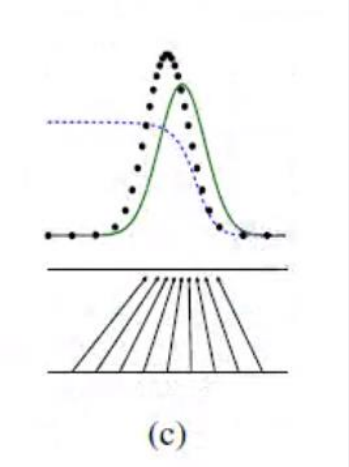
(b) w wewnętrznej pętli algorytmu D jest trenowany do rozróżniania próbek z danych, zbiegając do $D^*(x) = \frac{p_{data}(x)}{p_{data}(x)+p_g(x)}$



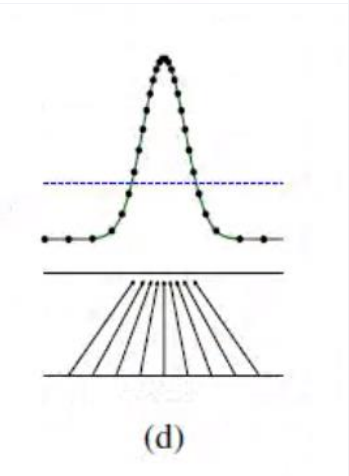
Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., 2014. Generative adversarial nets. *Advances in neural information processing systems*, *27*.

Generatywne sieci współzawodniczące

(c) po aktualizacji G, gradient D kieruje $G(z)$ do regionów, które z większym prawdopodobieństwem zostaną zaklasyfikowane jako dane



(d) po kilku epokach, jeśli G i D mają wystarczającą pojemność, osiągną punkt, w którym nie mogą się poprawić, ponieważ $p_g = p_{data}$



Dyskryminator nie jest w stanie rozróżnić tych dwóch rozkładów, tzn.

$$D(x) = \frac{1}{2}$$

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., 2014. Generative adversarial nets. *Advances in neural information processing systems*, *27*.

Deep Convolutional GAN (DCGAN)

Uczenie GANów zadanie trudne

- proces uczenia bardzo niestabilny
- zarówno G, jak i D, konkurują ze sobą w grze o sumie zerowej (mini-max) ➔ ulepszenie jednego modelu kosztem drugiego

Najlepsze praktyki

- *Downsample* ➔ *Strided Convolutions* (nie używaj *pooling layers*)
- Upsample ➔ *Strided Convolutions* (użyj *transpose convolutional layer*)
- nie używaj standardowego ReLU (użyj LeakyReLU)
- żyj funkcji *Batch Normalization* (np. znormalizuj wyjścia warstw po aktywacji).
- użyj *Gaussian Weight Initialization* (np. średnia 0,0 i odchyleniem standardowym 0,02)
- użyj funkcji *Adam Stochastic Gradient Descent* (np. z szybkością uczenia 0,0002 i parametrem beta1 0,5)
- przeskaluj obrazy do zakresu [-1,1] (np. użyj funkcji *Tanh* na wyjściu generatora)

Podsumowanie

- GAN-y to technika głębokiego uczenia służąca do trenowania modeli generatywnych zdolnych do syntezy obrazów wysokiej jakości
- Uczenie GANów jest z natury ***niestabilne*** i podatne na błędy, które można przezwyciężyć poprzez zastosowanie najlepszych praktyk
- Modele generatora i dyskryminatora używane w architekturze GAN można zdefiniować w prosty sposób i bezpośrednio w bibliotece uczenia głębokiego *Keras (https://keras.io)*
- Model dyskryminatora jest trenowany tak jak każdy inny model uczenia głębokiego do klasyfikacji binarnej
- Model generatora jest trenowany za pośrednictwem modelu dyskryminatora w architekturze modelu złożonego
- Zaawansowane GANy, takie jak skalowanie modeli i ich stopniowy wzrost, pozwalają na generowanie większych obrazów, umożliwiają generowanie większych obrazów o wyższej jakości

Wykorzystanie sieci neuronowych w optymalizacji grafiki i animacji

Dr inż. Marek Woda

Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Rozwoju Regionalnego Program Operacyjny Polska Cyfrowa na lata 2014-2020,
Oś Priorytetowa nr 3 "Cyfrowe kompetencje społeczeństwa" Działanie nr 3.2 "Innowacyjne rozwiązania na rzecz aktywizacji cyfrowej"
Tytuł projektu: „Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI Tech)”
Porozumienie nr POPC.03.02.00-00-0001/20-00