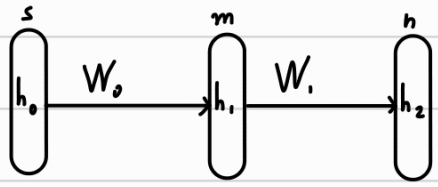
# Ideas to solve the instability issuses in predictive coding networks

Luca Pinchetti - 22/07/2024

! • GENERAL NOTE I DIDN'T KNOW WHERE TO PUT:
In the following document, I used the same network structure for both bp and pc, this means that the term "$h_i$" has a different meaning, as in bp is the activation of a layer, while in pc it is its state. Furthermore, it is quite implicit at the beginning, but in pc we are requiring that $V[h_i] = V[p_i]$

SETUP: Linear network with 2 layers (no bias + no act fn), gaussian sampling for weights.




$h_0: s^1$ $h_1: m^1$ $h_2: n^1$;

$W_0: m^s$ $W_1: n^m$

NOTATION: a $n \times m$ matrix $W$ is defined as $W: n^m$ $= n$ [ ] ($m$), such that a vector is $x: m^1$ and $y = W \cdot x: n^m \cdot m^1 = n^1$ (input dim: $m$, output dim: $n$)

BACKPROP: (Xavier et al., 2010)

① $V[h_0] = V[h_1] = V[h_2]$ $\overset{=d}{}$ ② $V\left[\frac{\partial F}{\partial h_0}\right] = V\left[\frac{\partial F}{\partial h_1}\right] = V\left[\frac{\partial F}{\partial h_2}\right] = k$

// my understanding is that $V[x]$ means all the elements of $x$ have the same variance $V[x]$

// ①②: the intuition is that we want constant "scale of information" through the network, so no exploding or vanishing activations/gradients

- since $E[h_i] = 0$; $E[W_i] = 0$ and $h_1 = W_0 h_0 \Rightarrow V[W_0 h_0] = V[h_1]$

then $V[W_0] = V[h_1] / sV[h_0] = \frac{1}{s} \Rightarrow \sigma_{W_0} = \frac{1}{\sqrt{s}}$ which is the standard initialization

($1 \Rightarrow =$)

using that $V[Ax] = nV[A]V[x]$ with $A: b^a$

and similarly, $V[W_0] = \frac{1}{m}$. Then, we can do a similar analysis for the gradients.

PREDICTIVE CODING

- same analysis holds for forward pass, having that if $V[W_i] = \frac{1}{d_i}$ then $V[h_i] = 1$, with $W: d_{i+1}^{d_i}$, which satisfies ①
- HOWEVER, this does NOT seem true for the gradients, and ② is not satisfied.

ANALYSIS (for $h_1$): $F = \frac{1}{2}(W_0 h_0 - h_1)^2 + \frac{1}{2}(W_1 h_1 - h_2)^2 \Rightarrow \frac{\partial F}{\partial h_1} = h_1 - W_0 h_0 + W_1^T(W_1 h_1 - h_2)$

thus ① $V\left[\frac{\partial F}{\partial h_1}\right] = V[W_0 h_0] + V[(W_1^T W_1 + I) h_1] + V[W_1^T h_2]$ (A: $*$, B)

$*$: using $V[X+Y] = V[X] + V[Y] + 2Cov[X,Y]$

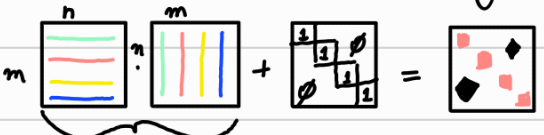
// I'm not 100% sure about $Cov[A,B] = 0$ since $W_1$ appears in both, but the final formula is correct and since we multiply $W_1$ by $h_1$ in A and by $h_2$ in B, they should be completely random.

$= V[h_1] + (m-1)n V^2[W_1] V[h_1] + V[h_1]\left(2nV[W_1]^2 + 1 + 2nV[W_1] + n^2 V^2[W_1]\right) + nV[W_1]V[h_2]$

(m-1 ◆, 1 ■)

... is true only if we are sampling from a Normal distribution.

- Explanation for ...: analysis of $W_1^T W_1 + I$




with $E[◆] = 0$, $V[◆] = nV[W_1]^2$

$E[■] = nV[W_1] + 1$, $V[■] = 2nV[W_1]^2$ // using ■ = □ + 1

1) Each off-diagonal element is obtained as $◆ = \sum_{i=1}^{n} x_i y_i$ where $x_i \in W_1^T$ and $y_i \in W_1$ are different elements of $W_1$ for all $i$, so they are independent $\Rightarrow V[◆] = nV[W_1]^2$, $E[◆] = 0$

*: using $V[xy] = V[x]V[y] + V[x]E[y]^2 + V[y]E[x]^2$ and $E[xy] = E[x]E[y]$ if $x, y$ independent

2) Each diagonal element is obtained as $□ = \sum_{i=1}^{n} x_i^2$ since we pick the same element in both $W_1^T$ and $W_1$. Since $x_i$ is Gaussian r.v. we have $E[x_i^2] = V[x] - E[x]^2 = V[W_1] \Rightarrow$ $E[□] = nV[W_1]$ and $V[□] = 2nV[W_1]^2$ (check online for this)

Luca Parolari 22/07/2024

thus we compute $V[Ah_1]$ as ... using $*$, with $A = W_1^T W_1 + I$.

Having $V\left[\frac{\partial F}{\partial h_1}\right]$, let's try to satisfy $V[h_1] = 1$ and $V\left[\frac{\partial F}{\partial h_1}\right] = 1$

formally I required all of them to be equal to constants j and k, but let's just use 1.

$V[h_i] = 1 \Rightarrow V[W_i] = \frac{1}{d_i}$ as shown initially. Let's substitute:

$$V\left[\frac{\partial F}{\partial h_1}\right] = 1 + (m-1)n\frac{1}{m^2}\cdot 1 + 2n\frac{1}{m^2} + 1 + n^2\frac{1}{m^2} + 2n\cdot\frac{1}{m} + n\cdot\frac{1}{m}\cdot 1 = 2 + \frac{(m-1)n}{m^2} + \frac{2n}{m^2} + \frac{n^2}{m^2} + 3\frac{n}{m}$$

$$= 2 + \frac{n(m+1)}{m^2} + \frac{3n}{m} + \frac{n^2}{m^2} \neq 1$$ // This formula has been verified numerically.

- $E[\blacksquare]$ and $V[\blacksquare]$ play a major role in the formula above. If somehow we assume we manage to "break the symmetry" of $W_1^T \cdot W_1$, we have that $E[\blacksquare] = 1$ and $V[\blacksquare] = n\,V[W_1]^2$. Then $V\left[\frac{\partial F}{\partial h_1}\right]$ becomes:

$$\tilde{V}\left[\frac{\partial f}{\partial h_1}\right] = 1 + (m-1)\frac{n}{m^2} + \frac{n}{m^2} + 1 + \frac{n}{m} = 2\left(1 + \frac{n}{m}\right) \neq 1$$

which looks better than the above.

and the original (w/out substitution) becomes:

$$\tilde{V}\left[\frac{\partial f}{\partial h_1}\right] = V[h_1] + (m-1)nV[W_1]V[h_1] + V[h_1]\left(nV[W_1]^2 + 1\right) + nV[W_1]V[h_2]$$

- QUESTION: can an activation function mitigate such symmetry? Now we don't have any, using one should lead to something like $W_1^T \sigma(W_1 h_1)$ instead of $W_1^T W_1 h_1$. Let's briefly consider having

$$F = \frac{1}{2}(W_0 h_0 - h_1)^2 + \frac{1}{2}\left[\sigma(W_1 h_1) - h_2\right]^2 \Rightarrow \frac{\partial f}{\partial h_1} = h_1 - W_0 h_0 + W_1^T\left[\left(\sigma(W_1 h_1) - h_2\right) \odot \sigma'(W_1 h_1)\right]$$

$$= h_1 - W_0 h_0 + W_1^T \sigma(W_1 h_1)\sigma'(W_1 h_1) - W_1^T h_2 \odot \sigma'(W_1 h_1)$$

- QUESTION: is this enough for independence?

Luca Pinchetti
22/07/2024

LET'S GO BACK TO THE LINEAR CASE.

- QUESTION: can we find different initialisations/energy functions that satisfy (1) & (2)?

  my ANSWER: (1) $\Rightarrow V[h_i]=V[h_{i-1}]=V[W_i h_i] \Rightarrow V[h_i]=d_i V[W_i]V[h_i]$, with $W_i: d_{i+1}^{d_i}$

  $\Rightarrow 1=d_i V[W_i]\cdot 1 \Rightarrow V[W_i]=\frac{1}{d_i}$ (which we know doesn't work)

  // is this correct? - what about having an act fn?

  $\Rightarrow$ the initialisation is forced

  // note that in Xavier 2010 the weight are initialised as $\frac{1}{k d_i}$. Not sure where k comes from, or if there should be any in PC, but it wouldn't change the result.

  $\Rightarrow$ we can try to act on the energy:

Let's assume $\frac{\partial f}{\partial h_1}=\alpha\left(h_1-W_0 h_0\right)+\beta\left(W_1^T\left(W_1 h_1-h_2\right)\right)$, as it is the only reasonable modification of the energy fn I could think of, with $\alpha=f(n)$ and $\beta=f(n)$. Then:

$$V\left[\frac{\partial f}{\partial h_1}\right]=V\left[\alpha W_0 h_0\right]+V\left[\beta\left(W_1^T W_1+\alpha I\right)h_1-\beta W_1^T h_2\right]=\alpha^2 V[h_1]+\beta^2 n\, V[W_1]V[h_2]+\beta^2(m-1)\,n\,V[W_1]^2 V[h_1]+$$
$$+\beta^2\left(2nV[W_1]^2+\alpha^2+n^2V^2[W_1]+2\alpha nV[W_1]\right)V[h_1]$$

$E[\blacksquare]=nV[W_i]+\alpha$

// ____ indicates terms that disapper when we "break the symmetry".

- if $V[h_1]=1 \Rightarrow V[W_1]=\frac{1}{m}$

$$\Rightarrow V\left[\frac{\partial f}{\partial h_1}\right]=\alpha^2+\beta^2\frac{n}{m}+\beta^2(m-1)\frac{n}{m^2}+\beta^2\frac{2n}{m^2}+\beta^2\alpha^2+\beta^2\frac{n^2}{m^2}+2\alpha\beta^2\frac{n}{m}$$

--- terms imply that $\alpha$ and $\beta$ must be constants $\Rightarrow$ also this doesn't work.

SO: we CAN'T satisfy both (1) and (2)

// are my conclusions correct?

NOTES: • $\alpha=\frac{1}{m}$ and $\beta=\frac{1}{n}$ would guarantee $V\left[\frac{\partial f}{\partial h_1}\right]<k$ (so no exploding grads, but still vanishing)

- if we "break the symmetry", with $E[\blacksquare]=\alpha$ and $V[\blacksquare]=n\,V[W_1]^2$, then

$$V\left[\frac{\partial f}{\partial h_1}\right]=\alpha^2+\alpha^2\beta^2+\beta^2\frac{n}{m}+\beta^2\frac{n}{m}=\alpha^2\left(1+\beta^2\right)+2\beta^2\frac{n}{m}$$

$\Rightarrow$ still no solution (as $\alpha$, $\beta$ must be constant)

SO WE NEED TO FIND DIFFERENT CONSTRAINTS THAT GUARANTEE STABLE TRAINING!!!

Luca Pinchetti 22/03/2024

- QUESTION: can we rewrite the constraints as:

(1) $V[h_{i+1}] = V[\overbrace{W_i h_i}^{y_{i+1}}]$ // activation and state have the same V

(2) $V\left[\frac{\partial F}{\partial h_i}\right] = V[h_i]$ // state and its error have the same V.

(3) $d_i = d_j \Rightarrow a < V[h_i] = V[h_j] < b \quad , a,b \in \mathbb{R}^+$

// if two layers have the same size, they must have the same variance, this is to write that the variance is a property of the layer and it doesn't depend on its position in the network (maybe there are better ways to write this).

? • do they achieve analogous training stability?

INTUITION: we don't need the variance to be GLOBALLY the same, but just to be LOCALLY stable: (1) guarantees that the incoming activation V matches the state V, such that they both have the same "scale"; (2) ensures that the same is valid for the gradient (so that the error is proportional to state, so it can't vanish or explode); finally (3) ensures V of a layer is bounded and non-vanishing.

Are these constraint correct? Am I missing something from Xavier 2010?

- QUESTION: can we satisfy these constraints?

my ANSWER: be $V[h_i] = \frac{1}{\sqrt{d_i}}$ // satisfies (3)

then $\Rightarrow d_i V[W_i] \cdot V[h_i] = V[h_{i+1}] \Rightarrow V[W_i] = \frac{\sqrt{d_i}}{\sqrt{d_{i+1} \cdot d_i}} = \frac{1}{\sqrt{d_{i+1} \cdot d_i}}$ // satisfies (1)

we have that: $V\left[\frac{\partial F}{\partial h_1}\right] = \alpha^2 \frac{1}{\sqrt{m}} + \beta^2 n \cdot \frac{1}{\sqrt{m \cdot n}} \cdot \frac{1}{\sqrt{n}} + \beta^2 (m-1) n \cdot \frac{1}{m \cdot n} \cdot \frac{1}{\sqrt{m}} + 2\beta^2 n \frac{1}{m \cdot n} \cdot \frac{1}{\sqrt{m}} + \frac{\beta^2 \alpha^2}{\sqrt{m}} + \frac{\beta^2}{\sqrt{m}} \frac{n^2}{n \cdot m} + 2\alpha\beta^2 \frac{n}{\sqrt{m} \sqrt{m \cdot n}}$

$= \frac{1}{\sqrt{m}} \left( \alpha^2 + \beta^2 + \beta^2 \frac{(m+1)}{m} + \beta^2 \alpha^2 + \beta^2 \frac{n}{m} + 2\alpha\beta^2 \sqrt{n} \right) \neq V[h_1] = \frac{1}{\sqrt{m}}$

Luca Pinchetti
22/07/2024

Thus it is NOT a solution.

HOWEVER, if we "break the symmetry": (remove ___ from above)

$$\tilde{V}\left[\frac{\partial F}{\partial h_1}\right] = \frac{1}{\sqrt{m}}\left(\alpha^2 + 2\beta^2 + \beta^2\alpha^2\right) = V[h_1] \text{ if } \alpha^2 + 2\beta^2 + \beta^2\alpha^2 = 1 \text{ !!}$$

// note that there may be an error in the derivation, since in the numerical solution I have that $\alpha = \beta = \frac{1}{2}$ works (however the numerical solution uses a different setup,) so the above could be right.

SO WE (ALMOST) FOUND A SOLUTION!

↳ since for example if $d_i = d_{i+1} \Rightarrow \alpha = \beta \Rightarrow 3\alpha^2 + \alpha^4 = 1$

$\Rightarrow \alpha^2 = \frac{-3 \pm \sqrt{9+4}}{2}$ is clearly not a realistic solution

$\Rightarrow$ there's likely a small error somewhere.
(but let's ignore it for a second)

QUESTION: How do we "break the symmetry"? 3 possibilities:

- using an activation function? (to be tested)
- using a separate set of weights E for the backward pass instead of $W^T$ (see Neural Generative Coding by Ororbia)?
- Using complex numbers!!

Apparently if $x \in \mathbb{C} \sim N(\emptyset, K + Kj)$ (not sure how to write "sampled from a normal distribution with $\emptyset$ mean K variance for both real and imaginary part)

then $\Rightarrow E[x^2] = \emptyset$ which we can use to have $E[\square] = \emptyset$ so to "break the symmetry".

Luca Pinchetti
22/07/2024

In particular (numerically tested, check py notebook), using the following satisfies (1), (2) and (3):

- $V[h_i] = \frac{1}{\sqrt{d_i}}$, $h_i : d_i^{\,1}$

- $V[W_i] = \frac{1}{2\sqrt{d_i \cdot d_{i+1}}}$, $W_i : d_i^{\,d_{i+1}}$

- $\alpha = \beta = \frac{1}{2}$

.... : factors probably due to the complex numbers

// All of these was done assuming normal distributions, slightly different results may be true for uniform distributions.

// and of course this holds only for linear networks.

DISCLAIMER: everything related to complex numbers was found via trial and error, so not sure the explanation I gave are correct.

QUESTION: is this solution unique? Does it apply to actual networks? can we get rid of complex numbers?

---

DISCUSSION:

- Xavier 2010 requires that $V[h_i]$ (and thus $V\left[\frac{\partial f}{\partial h_i}\right]$) $= 1$ (or a constant, but it shouldn't matter.), this implies that the "energy" of a layer, defined as $E = \frac{1}{2}\sum_{j=1}^{d_i} \left(h_i - W_{i-1}h_{i-1}\right)_j^2$, scales with the number of neurons in the last layer.
  In particular, $V[E] = d_i V^2[h_i]$ (again, assuming gaussian r.v.s), which explains (finally!!) why we use MEAN SQUARE ERROR in the last layer as loss $\mathcal{L} = \frac{1}{2d_i}\sum_{j=1}^{d_i} \left(h_i - W_{i-1}h_{i-1}\right)_j^2$ (instead of just square error), since

$$V[\mathcal{L}] = \frac{1}{d_i} V[E] = V^2[h_i]$$

i.e. the total loss doesn't depend on the size of the layer.

Luca Pinchetti
22/07/2024

- Having $V[h_i] = \frac{1}{\sqrt{d_i}}$ (and I guess $\alpha = \beta = \frac{1}{2+\frac{1}{0}}$ for complex numbers)

achieves the same result, since: $V[E] = d_i V^2[h_i] = d_i \cdot \left(\frac{1}{\sqrt{d_i}}\right)^2 = 1$

and we don't need the "mean" of the square error, justifying the difference between the loss function of a bp and a pc trained network (i.e., if we assume $V[h_i] = \frac{1}{\sqrt{d_i}}$, then both bp and pc use SQUARE ERROR loss, without any MEAN). ↳ finally!!

- We are effectively scaling the energy function, since we are going from $V[E_i] = d_i$ to $V[E_i] = 1$. Is this the solution to the "we need a custom optimizer to rescale the energy function" problem we all have been talking about for ~4 year?

- This does intuitively make sense, as each layer now has a fixed energy magnitude (i.e., variance) independently from the number of neurons; that is, we do not generate new energy from thin air by simply adding new neurons to a layer, but, instead, existing neurons have to share the fixed total energy. This seems reasonable if we assume a system with a finite amount of energy like the brain, or, may be, analog hardware.

- IMPORTANT: using the conditions above $\left(V[h_i] = \frac{1}{\sqrt{d_i}}, V[W_i] = \frac{1}{2\sqrt{d_i d_{i+1}}}, \alpha = \beta = \frac{1}{2}\right)$

also satisfies a fourth constraint that we may find desiderable:

(4) pre and post synaptic errors have the same variance, namely, given $e_{1<} = (W_0 h_0 - h_1)^2$ and $e_{1>} = (W_1 h_1 - h_2)^2$, we have that $V[e_{1<}] = V[e_{1>}]$.
(see latest sections of the py notebook)

Luca Pinchetti
22/03/2024

# UPDATE 23/07/2024

- Using a different matrix for the error connections ($E_i$ instead of $W_i^T$) seems to solve all the problems since it sets $V[\square] = V[\blacksquare]$ and $E[\square] = E[\blacksquare]$.
  Furthermore it works even with $V[h_i] = 1$ and $V[W_i] = \frac{1}{d_i}$, using $V[E_i] = \frac{1}{d_{i+1}}$ which is the original xavier method (but also with the new defined variances).

=> the question becomes: "complex numbers or asymmetric weights"? They both require doubling the amount of parameters, so the question is which one trains faster, or other advantages?

=> I found this overall very interesting as PC was proposed as a bio inspired learning algorithm, and here we proved that it CANNOT work with symmetric backward weights, which is perfectly inline with the asymmetries found in the human brain.

- Using $V[h_i] = \frac{1}{k\sqrt{d_i}}$, k doesn't seem to be important, since it can always be balanced by a coefficient in the energy function. Is there any difference between different K values? How to choose?

---

# UPDATE 30/07/2024

- FURTHER ANALYSIS OF REC-LRA

  - as mentioned, in rec-LRA (and NGC which can be seen as a derivation of it), we introduce an error matrix $E_i$ to backpropagate the error. Further analysis of the algorithm suggests that the goal of $E_i$ is simply to behave as $W^T$, since $\nabla E = \nabla W^T$. The question than becomes how impactful is the

Luca Pinchetti 22/07/2024

different initialisation of the two matrixes ($W$ and $E$). Experimentally it seems like the norm of the two matrices significantly grows with training, suggesting that the initial point is quite irrelevant. However, the gradient analysis proposed in this document refers to the initial training steps, during which $E$ and $W^T$ are actually different (but also random, so not very good at training the model perhaps).

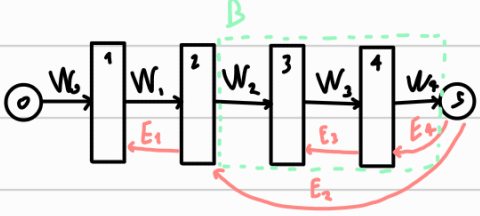
However, there (probably) are alternative ways we can define $\frac{\partial f}{\partial h_i}$ such that the error is propagated by an auxiliary error matrix $E$, such that $\frac{\partial F}{\partial E}$ is well defined:

1) $\tilde{F}_i = \frac{1}{2}\left(W_{i-1}h_{i-1} - h_i\right)^2 + \frac{1}{2}\left(h_i - E_i h_{i+1}\right)^2 \Rightarrow \frac{\partial \tilde{F}_i}{\partial h_i} = \left(h_i - \mu_i\right) + \left(h_i - E_i h_{i+1}\right)$

2) $\frac{\partial \tilde{F}_i}{\partial h_i} = \left(h_i - \mu_i\right) + E_{i+1}\left(\mu_{i+1} - h_{i+1}\right) \Rightarrow \frac{\partial \tilde{F}}{\partial E_{i+1}} = ?$ E <u>metalearning</u>?

- Analysis of $\nabla E$.

In recLRA, we have some "Error Skip Connections" as following:



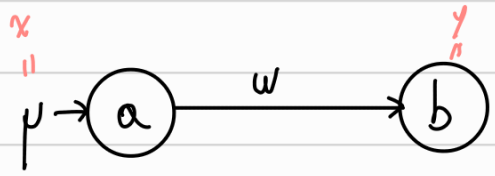

As just described, $\nabla E_i = \nabla W_i^T$ which basically implies that $E_1, E_3, E_4$ are simply $W_1^T, W_3^T, W_4^T$ and we may not even need them. But what about $E_2$? Clearly $E_2$ doesn't correspond to $W_2^T$. My intuition is that we can approximate B as a linear transformation represented by $\tilde{W}_2$ and assume that $\mu_5 = \tilde{W}_2 h_2$ and thus $\nabla E_2 = \nabla \tilde{W}_2^T$.

This is effectively a error approximation method. Can then refinement be achieved with inference? What about more complex approximations? (High order or NLPs or CNNs or other)?

However, this interpretation leads to many different implementations:

- is the error carried by $E_2$ complementary or alternative to the error carride from $h_3$ to $h_2$? (Could $E_2$ be an "initialisation value" for the error node?)
- is B actually representing $\tilde{W}_2$, or do we need a forward skip connection as well?

$x = -1$
$y = 1$
} Task: invert the input!




- for bp is trivial!

OBVIOUS SOLUTION: $w = -1$, HOWEVER, LET'S SET $w_0 = 1$

- for pc it doesn't work at all

=> UNLESS: inference is not performed to convergence, but stopped way before !!!

It explains: why a low T works better !!

If convergence is achieved (assuming a minimum of noise) (but only in this particular example noise is necessary)

w will keep increasing until it diverges !! => exactly what we experienced.

(≈ "summer vacation")

=> might explain the success of iPC: trains giving more emphasis to the forward info.

=> backward inference? very likely affected, might not be solution tho.

Brief Analysis:

$2f = (a - x)^2 + (b - wa)^2$, if $x = -1, y = 1$

$\Rightarrow 2f = (a+1)^2 + (1 - wa)^2$

$\Rightarrow \frac{\partial f}{\partial a} = a + 1 - w(1 - wa) = a + 1 - w + w^2 a$

$\frac{\partial f}{\partial a^*} = 0 \Rightarrow a^* = \frac{w-1}{w^2+1}$ } if $w > 1$, then $a^* > 0$

$$\Rightarrow \frac{\partial f}{\partial w} = -\vec{a}\left(1 - w\vec{a}\right) = -\vec{a}\cdot\vec{\varepsilon}^{*} \qquad \vec{\varepsilon}^{*} = 1 - w\vec{a}^{*} = 1 - \frac{w\,(w-1)}{w^2+1}$$

$$= -\frac{w-1}{w^2+1}\cdot\frac{w+1}{w^2+1} = -\frac{w^2-1}{w^2+1}$$

if $w > 1$, then $\frac{\partial f}{\partial w} < 0$

$$= \frac{w^2+1-w^2+w}{w^2+1} = \frac{w+1}{w^2+1}$$

w will keep increasing from the value of 1 for ever, instead of approaching the solution $w = -1$.

$$\frac{e^{i\pi w}-1}{e^{2i\pi w}+1}$$

$$(a+1)^2 + \left(1 - e^{i\pi w}a\right)^2$$

$$x = \frac{w-1}{w^2+1}$$